UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

BRUNO MOREIRA NABINGER

A Deep Learning Palpebral Fissure Segmentation Model in the Context of Computer User Monitoring

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Electrical Engineering

Advisor: Dr. Tiago Oliveira Weber

Porto Alegre
September 2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões Mendes
Vice-Reitora: Prof^{a}. Patrícia Helena Lucas Pranke
Pró-Reitora de Graduação: Prof. Cíntia Inês Boll
Diretor da Escola de Engenharia: Prof^{a}. Carla Schwengber ten Caten
Coordenador do Curso de Engenharia Elétrica: Prof. Ivan Müller
Bibliotecária-chefe da Escola de Engenharia: Rosane Beatriz Allegretti Borges

ABSTRACT

The intense use of computers and visual terminals is a daily practice for many people. As a consequence, there are frequent complaints of visual and non-visual symptoms, such as headaches and neck pain. These symptoms make up Computer Vision Syndrome and among the factors related to this syndrome are: the distance between the user and the screen, the number of hours of use of the equipment and the reduction in the blink rate, and also the number of incomplete blinks while using the device. Although some of these items can be controlled by ergonomic measures, controlling blinks and their efficiency is more complex. A considerable number of studies have looked at measuring blinks, but few have dealt with the presence of incomplete blinks. Conventional measurement techniques have limitations when it comes to detecting and analyzing the completeness of blinks, especially due to the different eye and blink characteristics of individuals, as well as the position and movement of the user. Segmenting the palpebral fissure can be a first step towards solving this problem, by characterizing individuals well regardless of these factors. This work investigates with the development of Deep Learning models to perform palpebral fissure segmentation in situations where the eyes cover a small region of the images, such as images from a computer webcam. The segmentation of the palpebral fissure can be a first step in solving this problem, characterizing individuals well regardless of these factors. Training, validation and test sets were generated based on the CelebAMask-HQ and Closed Eyes in the Wild datasets. Various machine learning techniques are used, resulting in a final trained model with a Dice Coefficient metric close to 0.90 for the test data, a result similar to that obtained by models trained with images in which the eye region occupies most of the image.

Keywords: Palpebral fissure. UNet. LinkNet. Computer Vision Syndrome. incomplete blink.

Modelo de aprendizagem de máquina profunda para segmentação da fissura palpebral no contexto do monitoramento de usuários de computador

RESUMO

A utilização intensa de computadores e terminais visuais é algo cotidiano para muitas pessoas. Como consequência, queixas com sintomas visuais e não visuais, como dores de cabeça e no pescoço, são frequentes. Esses sintomas compõem a Síndrome da visão de computador e entre os fatores relacionados a essa síndrome estão: a distância entre o usuário e a tela, o número de horas de uso do equipamento e a redução da taxa de piscadas, e, também, o número de piscadas incompletas, durante a utilização do dispositivo. Ainda que alguns desses itens possam ser controlados por medidas ergonômicas, o controle das piscadas e a eficiência dessas é mais complexo. Um número considerável de estudos abordou a medição de piscadas, porém, poucos trataram da presença de piscadas incompletas. As técnicas convencionais de medição apresentam limitações para detecção e análise completeza das piscadas, em especial devido as diferentes características de olhos e de piscadas dos indivíduos, e ainda, pela posição e movimentação do usuário. A segmentação da fissura palpebral pode ser um primeiro passo na resolução desse problema, caracterizando bem os indivíduos independentemente desses fatores. Este trabalho aborda o desenvolvimento de modelos de *Deep Learning* para realizar a segmentação de fissura palpebral em situações em que os olhos cobrem uma região pequena das imagens, como são as imagens de uma webcam de computador. Foram gerados conjuntos de treinamento, validação e teste com base nos conjuntos de dados *CelebAMask-HQ* e *Closed Eyes in the Wild*. São utilizadas diversas técnicas de aprendizado de máquina, resultando em um modelo final treinado com uma métrica Coeficiente Dice próxima a 0,90 para os dados de teste, resultado similar ao obtido por modelos treinados com imagens nas quais a região dos olhos ocupa a maior parte da imagem.

Palavras-chave: fissura palpebral, UNet, LinkNet, Síndrome da Visão do Computador, piscada incompleta.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
CCA	*Connected-component analysis*
CEW	*Closed Eyes in the Wild* (dataset)
CPU	*Central Processing Unit*
CNN	*Convolution Neural Network*
CSV	*Computer Vision Syndrome*
DED	*Dry Eye Disease*
DES	*Digital Eye Strain*
EAR	*Eye Aspect Ratio*
EARM	*Eye Aspect Ratio Mapping*
FCN	*Fully Convolution Networks*
FPS	*Frames per second*
GPU	*Graphics Processing Unit*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
IOU	*Intersection over Union*
IPH	*Interpalpebral Height*
POH	*Palpebral Opening Height*
RAM	*Random-Access Memory*
SVM	*Support Vector Machine*
VDT	*Visual display-terminal*

CONTENTS

1 INTRODUCTION

The use of computers and others visual display-terminals (VTD) is a daily practice for a large part of the world's population. As a result, asthenopic symptoms associated with Computer Vision Syndrome (CVS) are common: visual complaints, ocular disorders, eye fatigue, eyestrain, dry eyes, and other visual diseases. American Optometric Association (1997) first presented CVS definition as "the complex of eye and vision problems related to near work experienced during computer use". It can also be referred to as Digital Eye Strain (DES) or VDT syndrome, emphasizing that it is not only associated to computer use. Yan et al. (2008) indicates that headache and neck pain are commonly observed CVS symptoms.

The emergency health situation caused by COVID-19 (SARS-CoV-2), declared a pandemic disease by the World Health Organization in 2020, has given this topic new relevance. The lockdown and remote working measures adopted in several countries have abruptly changed the behaviors and lifestyles of the world population in general. Barros et al. (2022) found a significant association in screen use for college lecturers in Brazil giving remote classes, and the occurrence of asthenopia. This was especially true in groups with longer screen time. Improper body posturing, not taking breaks, long-hour duration of VDT use, and short-distance screen were all aspects associated with increased odds of CVS by studies reviewed by Anbesu and Lema (2023).

While environment design measures can be taken prior to engaging in the computer activities to reduce issues (DAIN; MCCARTHY; CHAN-LING, 1988), effects in blinking pattern are not so simply addressed. Blinks may be partially inhibited when subjects are engaged in a visual tracking task. According to Kennard and Smyth (1963), blinking and visual search functions have been shown to be mutually inhibitory. Thus, during a demanding visual task, blinking frequency is reduced and, conversely, it occurs when the task is interrupted or ceases. Rosenfield (2011), in a review of CSV causes, observed that while vergence[1] and accommodation [2] responses to VDT appear to be similar to those found for printed materials, dry eye symptoms are greater during computer usage. The reason is considered to be probably because of the decrease in blink rate and blink amplitude, as well as an increase of corneal exposure due to the monitor frequently being positioned in primary gaze.

While studying the interaction between blinking and vertical following move-

[1] movement in opposite directions of eyes to obtain or maintain single binocular vision

[2] process by which the eye changes focus from distant to near objects

ments of eyes and lid, Kennard and Smyth (1963) noted the presence of "miniature", partial blinks, during a demanding visual tracking task. According to Portello, Rosenfield and Chu (2013), while CVS symptoms are associated with a reduced blink rate, blink completeness can be equally significant. Hirota et al. (2013) concluded that even if the total blink rate decreases, tear film remains stable if almost all blinks are complete. McMonnies (2021) highlights that measures and remediation focus in only total blink rate have limited usefulness in the diagnosis and treatment of blink inefficiency-related ocular surface exposure, dry eye symptoms and ocular surface disease.

Reliably measuring blinks is thus needed, specially because blink patterns change with VDT use and the time spent in the task. Manually analyzing long videos (couple of hours) is a cumbersome process, and many studies have verified blink patterns for only a short period of time. Considering what was discussed before, investigating ways to measure eye closeness is relevant. There is no exact model for the determination of blinks and, in particular, incomplete blinks. Blinks are affected by many internal and external factors, as shown in Table 1.1.

Table 1.1: Some factors that affect blink

Internal factors affecting blink	**External factors affecting blink**
Age, Gender	Change in environment
Ocular surface exposure and damage;	Conversation
palpebral aperture size	Reading
Tear film break up time, rate of tear evaporation	**Computer and e-reader use**
Visual acuity	Sleep
Mental/Muscular fatigue and tension	Time of day
Contact lens wear and drug interactions	Illumination
Mental disorders and emotional state	Temperature and humidity
Concentration and cognitive processes	Air movement
Ocular saccades and shifting gaze (focal distance)	Noise
Awareness of measurement	...

Source: Adapted from Rodriguez et al. (2018).

Zheng et al. (2022a) have used a special type of deep learning fully convolutional network, the UNet (RONNEBERGER; FISCHER; BROX, 2015), to detect incomplete blinks. This was done with high-resolution palpebral fissure images obtained with a Keratograph 5M. According to McMonnies (2021), however, blink performance during clinical assessment of blink efficiency is unlikely to be characteristic of or relevant to the blink inefficiency that develops and causes symptoms during patient’s various daily activities.

It is also difficult to measure blinks in normal situations (out of a laboratory or clinic). Cruz et al. (2011) indicate that using a commercial camera with a temporal resolution of 30 frames per second (FPS) may be employed to film the palpebral fissure to analyze spontaneous eye blink. This method is not completely objective, as blinks have a wide range of amplitudes, and the observer has to decide which upper lid movements should be considered a blink. When blinking, the lid does not come to the original position, for example (STERN; WALRATH; GOLDSTEIN, 1984). Differentiating a complete blink from an incomplete blink is especially challenging for humans. Fogelton and Benesova (2018) experienced this when annotating datasets obtained by cameras. Strictly speaking, a complete blink happens when the two eyelids touch. The authors have sometimes found it difficult to annotate this situation on videos recorded at 15 FPS, mostly because eyelashes may make not fully closed eyes look like closed eyes.

In this study, the segmentation of the palpebral fissure by the means of deep learning techniques is considered for situations where the eyes cover only a small region of the images, that corresponds to webcam images of computers and laptops, ultimately returning in each instance to a discussion of how these techniques may be used in the context of CVS and blink completeness analysis applications.

The solution of tasks involving data-driven computational intelligence presents several peculiarities. Cortacero, Fischer and Demiris (2019) highlight that the performances of the methods used in gaze direction estimation and blink detection tasks are significantly impacted by the database used. Deep learning models, for example, benefit from large datasets with many annotated examples of blinks. Using pretrained models can bootstrap deep learning applications through transfer learning. Furthermore, data augmentation and some architectures like UNet have been particularly successful strategies even when little data is available, as is the case of medical segmentation.

The analysis of the periocular region, of which the palpebral fissure is part, and eye closure/blink have many potential uses and have been the subject of many studies that are connected fields to this research. The iris is an individual characteristic that may be used for recognition (LOZEJ et al., 2018). Biometrics applications where the image of the iris does not have a good resolution may benefit of the segmentation of the palpebral fissure components for authentication processes (LUCIO et al., 2018). Eye closure and blink detection, in turn, may also be of interest for authentication of users, as in face anti-spoofing applications Pan et al. (2007), driver-fatigue alert systems (ALPARSLAN; ALPARSLAN; BURLICK, 2020) (HUDA; TOLLE; UTAMININGRUM, 2020), gaze estimation methods (CORTACERO; FISCHER; DEMIRIS, 2019), making realistic animations (TRUTOIU et al., 2011), engagement with content (RANTI et al., 2020), and the prevention and study of computer user related visual problems (YIN et al., 2022), Su et al. (2018), Zheng et al. (2022a). A recent case of the use of blink detection is the detection of fake face generated videos (LI; CHANG; LYU, 2018). The reduced number of face images with closed eyes compared to open eyes and the complex patterns of blink are a challenge for the techniques creating these videos.

In the chapter 2 Theoretical Background, physiological, ergonomic, and technical information are introduced to the reader to provide the necessary context for the following chapters. This includes information about the palpebral fissure, eye blink, and ergonomic and ophthalmic aspects of computer use. These are useful from the perspective of enhancing the segmentation with geometrical constraints (detecting faulty segmentations automatically) and of using palpebral fissures for blink completeness detection in videos. Elements of machine and deep learning are also reviewed. A literature review is then presented in chapter 3 Related work discussing the benefits of the proposed palpebral fissure segmentation approach. The chapter 4 Methods and methodology addresses the datasets used and the choices impacting data preparation, as well as models training, validation, testing, and display of relevant metrics. The results obtained will be presented and discussed in the chapter 5 Results and Discussion. Finally, in the chapter 6 Conclusion, a conclusion is made based on the results obtained and the perspectives of future works.

1.1 Objective

To develop a segmentation model of the palpebral fissure region using images where the eyes cover only a small region of the image, like the ones that could be obtained by a computer webcam.

1.2 Specific objectives

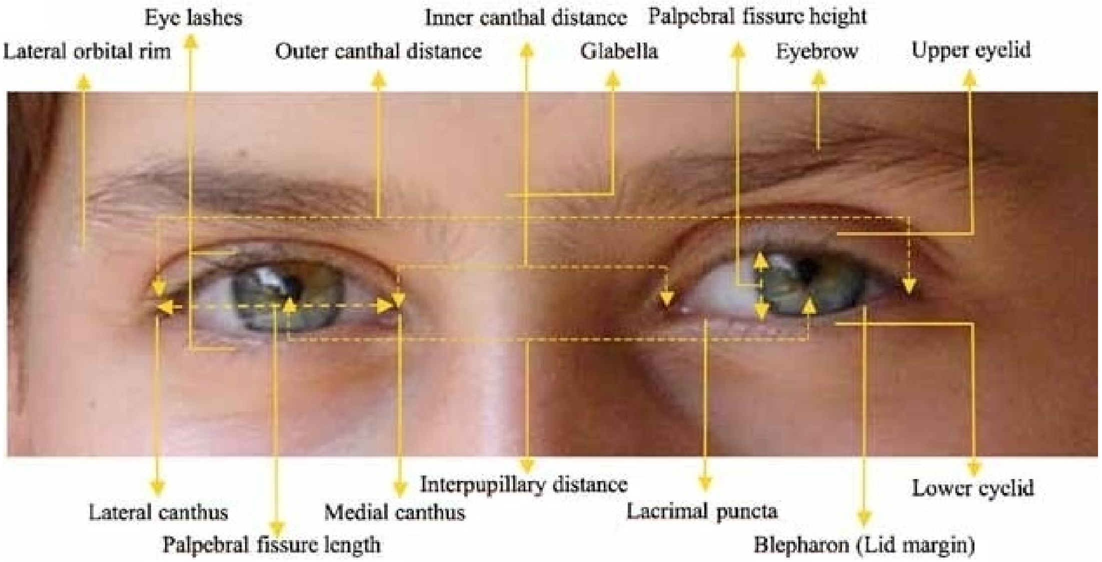
- to address how the segmentation of the palpebral fissure region can be used for analysis of blinks and their completeness, to aid in the diagnosis of CVS, from webcam images;
- approach how deep learning and post-processing techniques can be employed for the generation of palpebral fissure segmentation models;
- compare different architectures and topologies of models.

2 THEORETICAL BACKGROUND

2.1 Palpebral fissure, blink and computer use

The Figure 2.1 shows a periorbital image with a number of landmarks indicated. Major periorbital features commonly measured are the outer canthal distance (distance between the lateral canthi of the eyes), the interpupillary distance, the inner canthal distance (distance between the medical canthi of the eyes) and the palpebral fissure length (distance between the inner and outer canthi of the eye) (HALL et al., 2009). The Lacrimal punctum (plural: puncta) corresponds to the external aperture of the tear duct system.

Figure 2.1: Periorbital anatomy and terminology



Source: Hall et al. (2009).

The palpebral fissure is the longitudinal opening between the eyelids, approximately of elliptical shape. It extends from the lateral canthus (outer canthus) to the medial canthus (inner canthus). The normal adult eyelids frame a palpebral fissure measuring about 8 to 11 mm vertically at the pupillary meridian by about 30 to 33 mm horizontally (FANTE, 2007) (PRAKALAPAKORN et al., 2023).

Many factors, like the size, slant, eyelid architecture, and ptosis (blepharoptosis, the "dropping or falling" of the upper eyelid), can contribute to configuration of the palpebral fissures (GRIPP et al., 2013). The eye fissure dimensions vary according to age, gender, and ethnicity (VASANTHAKUMAR; KUMAR; RAO, 2013). Physiological palpebral fissure asymmetry is common, with the ocular dominance known to have a relatively small yet significant effect on the palpebral fissure height (DOGANAY et al., 2017) and width (GRIPP et al., 2013). Furthermore, horizontal and vertical palpebral fissure dimensions change in downward gaze (READ et al., 2006).

2.1.1 Eyeblink

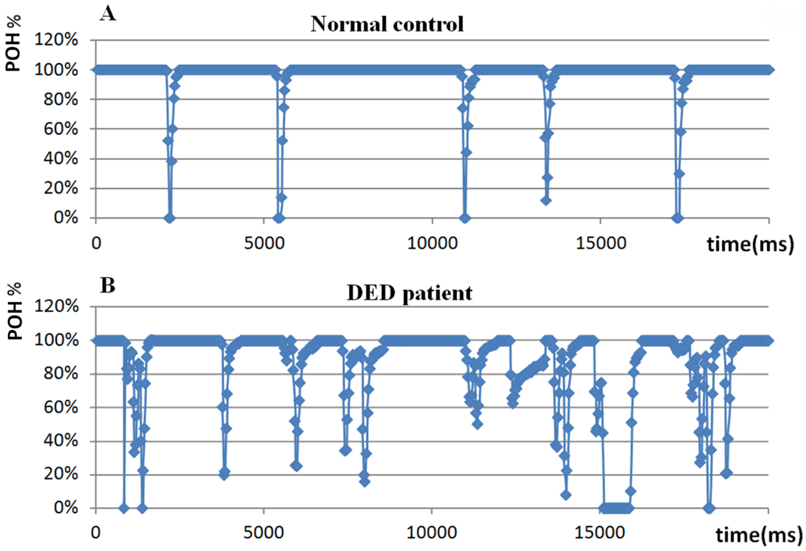
Eye closure can be associated to the different types of blink, i.e., endogenous blink, reflex blinks and voluntary blinks, and non-blink closures events, like sleep and microsleep. According to Stern, Walrath and Goldstein (1984), for blinks, the time from initiation of the eyelid movement to full eye closure takes in general less than 150 ms, contrasting with a time usually greater than 250 ms to close the eyes for non-blink closure. The authors also affirm that the full reopening time for blinks is about 100 to 200 ms, whereas for non-blink, the reopening seldom takes as much as 100 ms. Any closure for which the time to close is greater than 300 ms, the closure duration is greater than 1 and the time to reopen is less than 150 may not be relevant to blink analysis, although are relevant in the study of attention, alertness, and drowsiness.

When considering time aspects of the blink, for developing more robust state machines or recurrent neural networks, considering the opening and closing phases of eye blink may be relevant. Stern, Walrath and Goldstein (1984) states that a reasonable estimate for total time taken for lid closure in a blink is from 50 ms to 145 ms. The reopening takes longer than closing. It is relevant to note that the determination of blink completion (not to be confounded with the completeness of the blink) is based in the final quiescent (repose, inactivity) of the lid, as it may not return to the same position. A relatively steady level lasting some tens of milliseconds may be sustained at full closure before reopening.

The excessive exposure to visual display terminals has been associated with worsening dry eye symptoms, and possibly contributed to the increased incidence of dry eye syndrome (DED) in research conducted by Nutnicha et al. (2021) in Thailand.

Su et al. (2018) concluded that partial blinks, prolonged closed eyelid time, and short blink intervals were the three main characteristics observed in DED patients. Zheng et al. (2022a) demonstrated that incomplete blink frequency is correlated with DED symptoms and signs. Figure 2.2 shows the blink pattern of a normal control and a DED patient.

Figure 2.2: The blink pattern of a normal control and a DED patient



Palpebral opening height (the vertical distance between the central points of the upper and lower eyelid margins) percentage (POH %) was evaluated as the POH divided by the maximum POH opening observed during the recorded blinks.

Top: regular and symmetric blink pattern in normal controls; Bottom: irregular and partial blinks and asymmetric patterns for blink pattern of a DED patient.

Source: Su et al. (2018).

2.1.2 Ergonomic and ophthalmic aspects of computer use

Yan et al. (2008) indicate that eyes use significantly more muscles when focusing on objects at a near distance. Near work at the computer (viewing distance of less than 20 inches/ 50.8 cm) and long-hour use of the computer (3 h/day or more) are major factors in CSV. Authors state that a minimum of 20 inches for the distance between the user's eyes and the screen should be kept, as suggested by clinical optometrists. A comfortable horizontal distance for viewing the screen is usually around an arm's length. Distances of about 35 to 40 inches (88.9 to 101.6 cm) can produce fewer complaints of visual effort, since they allow users' eyes to relax (YAN et al., 2008).

Ankrum (1996) discusses the scientific evidence for the limits of the distance between the user and a monitor. The author points out that there is no scientific basis for a maximum distance limit. He recommends a minimum distance of at least 25 inches (about 63.5 cm), indicating that a slightly smaller distance is not bothersome for some people. Jaschinski-Kruza (1988) found less eye strain at a viewing distance of 100 cm than at 50 cm. Some national agencies have guides to the installation and operation of computers. For example, German Social Accident Insurance recommends that the distance between the eyes and the screen should be approximately 500 mm to 600 mm (Deutsche Gesetzliche Unfallversicherung, 2019).

While studying the relationship between dry eyes and video display terminals use, Tsubota and Nakamori (1993) have shown that the rate of tear evaporation increases as the ocular surface area exposed, which is a function of the palpebral fissure width, increases. Furthermore, the rate of the increase was found to be greater for larger surface areas. The authors have suggested that this was possible due to an instability of the tear film over a larger surface caused by the thinning mucin and lipid layers of the film.

Based on this study, Tsubota and Nakamori (1993) suggest that computer users lower the monitor and tilt the screen upward, as eyelids partially close when we look downward, reducing tear evaporation, which plays an important role in ocular fatigue after prolonged work at VDT. Yan et al. (2008) points out that computer users should adjust their computer monitors to a viewing angle of about 15° lower than the horizontal level. This angle is likely to reduce both visual discomfort (for example dry eyes), and musculoskeletal discomfort (like neck and back pain). The American Optometric Association (n.d.) promotes the 20/20/20 rule. It states that after 20 minutes of computer usage (VDT, in general), the user should look at a point 20 feet (about 6 m) distance for 20 seconds.

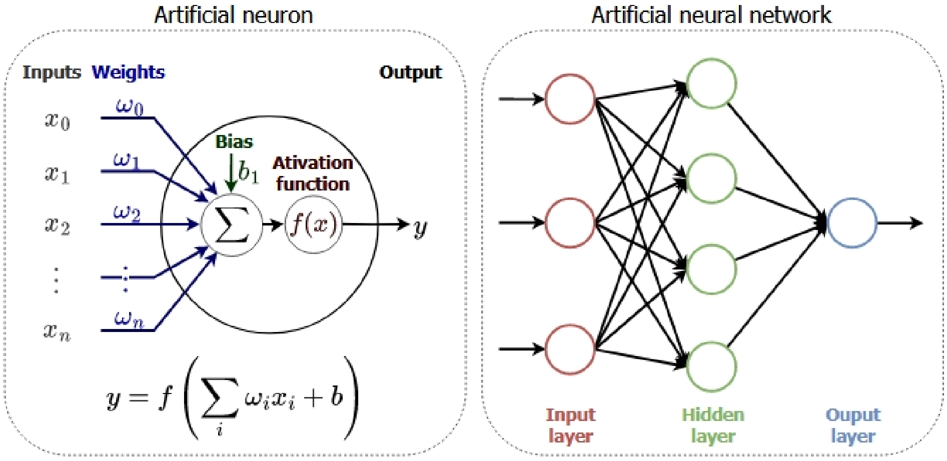
2.2 Machine Learning and Deep Learning

Machine learning is the study field related to allowing a computer model to "learn" and perform a task without been explicitly programmed for it, with the ability, for example, to deal with new data. Training a model means determining suitable values for all its non-predefined internal parameters (weights and bias, for an artificial neural network).

2.2.1 Artificial neural network (ANN)

An artificial neural network (ANN) is a machine learning model inspired in the human brain that consists of several interconnected neurons (arrangement of computational mathematical cells), with an input layers, at least one hidden layer and an output layer. Figure 2.3 displays one artificial neuron model and an example of ANN with only one hidden layer. A deep learning model is an ANN composed of multiple hidden layers.

Figure 2.3: Artificial neuron model and artificial neuron network



Source: Author.

There are many algorithms for training ANN, with a considerable number been based in the gradient descent method (LECUN et al., 1998a) (LECUN et al., 1998b), that is typically used to find the local minimum of a function.

2.2.2 Feature scaling techniques

The magnitude of features affects machine learning models like ANNs for various reasons. Scaling the data is relevant for algorithmic stability and prevents sensitivity to the scale of input features. Standardization corresponds to rescaling data so that it will have a mean of 0 and a standard deviation of 1. Normalization means collapsing the inputs range to a value between 0 and 1. In the case of 8 bits black-and-white images, it is usual to perform normalization by simply dividing the pixel intensity by 255.

2.2.3 Epoch, step, batch, and batch size

The number of epochs is the number of complete passes through the training dataset, being an integer value between *one* and infinity (theoretically). An epoch may contain many steps and batches. A batch tends to represent the distribution of the data better than a single input, with larger batches generally making for a better approximation, but tacking more time to be processed.

A model receives between 1 and all the training set samples before updating its internal parameters (i.e., a forward and backward pass known as "step"). This number is the batch size and when it is between 1 and the size of the training set (both extremes not included), the batch is called mini-batch. Common values include 32, 64, and 128.

2.2.4 Learning rate

The learning rate, or step size, in algorithms of optimization based on gradient descent, is a scalar value that multiplies the gradient, used to update the weights of the model. It effectively controls the speed of the learning process. If the learning rate is too small, the algorithm would take too long to converge (or get stuck in a local minimal that can be far from the actual minimal). If it is too large, the algorithm may fail to converge at all, by missing the minimal value and assuming a bouncing pattern.

The learning rate is also related to other techniques being employed and can change during training. While training Convolution Neural Networks (discussed in subsection 2.3.1 Convolutional neural network (CNN)) Krizhevsky, Sutskever and Hinton (2012) and Simonyan and Zisserman (2014) used 0.01 as a starting point, dividing this value by 10 when validation error rate stopped improving. In He et al. (2016), learning rate starts 0.1, also dividing by 10 when the error plateaus. Chollet (2017) informs that Inception V3 used 0.045, with a decay of rate 0.94 every 2 epochs.

2.2.5 Activation functions

Activation functions are part of the neurons of ANN, mapping the weighted sum of inputs to the output. They incorporate non-linearities to the network, allowing it to solve non-linearly separable problems. Two examples of activation functions are the sigmoid, and Rectified Linear Unit (ReLU), expressed by Equation 2.1. ReLU was one of the major factors in the success of deep learning model of Krizhevsky, Sutskever and Hinton (2012) for image recognition that has surpassed traditional computer vision approaches.

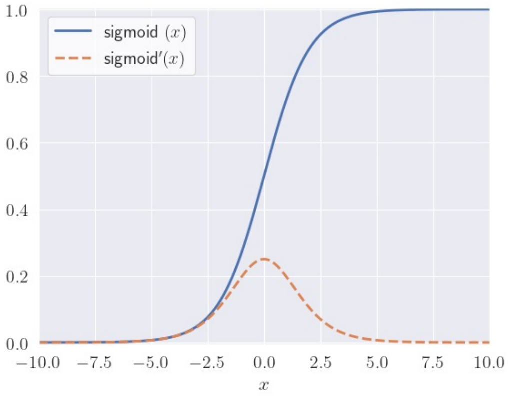
$$f(y_i) = \begin{cases} y_i, & \text{if} y_i > 0 \\ 0, & \text{if} y_i \leq 0 \end{cases} \tag{2.1}$$

The sigmoid function, shown in Figure 2.4 with its first derivative, is monotonically increasing, continuous everywhere, and also differentiable everywhere in its domain. Its definition is done in Equation 2.2:

$$\sigma = \frac{1}{1+\exp(-x)} \tag{2.2}$$

where x is the input, a real number. The output of sigmoid function is in the interval $(0;1)$.

Figure 2.4: Sigmoid and sigmoid's first derivative



.

Source: Author.

2.2.6 Imbalanced dataset – stratification

Some problems can exhibit a significant imbalance in the class distribution of the dataset. This is common in medical applications, where collecting samples can be difficult, and an especial condition is rarer than its absence. In these cases, it is important that each set (training, validation, and test) contain approximately the same percentage of samples from each class as the complete set. This is done by stratified sampling, a method that guarantees that the relative frequencies of classes are approximately preserved in sets.

2.3 Deep Learning

2.3.1 Convolutional neural network (CNN)

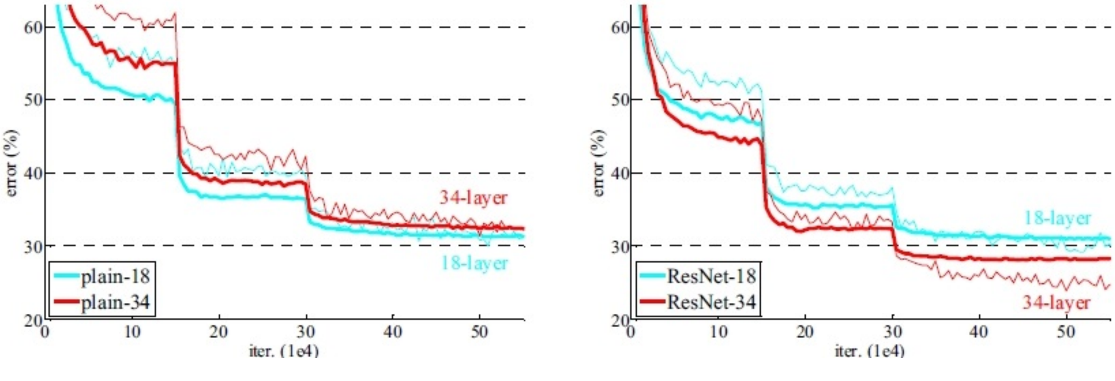
Convolutional neural networks (CNNs) are a special type of neural network that have the ability to learn a hierarchical representation of raw input data without relying on features obtained by digital filters in a preprocessing stage. They are based on convolutional layers, that perform feature extraction, and pooling operations, for spatial subsampling. The convolutional layers act as digital filters that learn to extract useful features from the data through training, a step that would normally be done in a preprocessing step for a traditional ANN and would require specific knowledge in the field of application. The pooling layers reduce the dimension of the data by combining the output of the previous layer, filtering out details.

In a CNN, the first layers learn low level features, for example how to detect lines and shapes. The last layers are capable of detecting complex structures, like contours of objects, animals, or faces.

2.3.1.1 ResNet

While training deep neural networks with increasingly more layers, the performance of the model stars to drop, as in Figure 2.5, in a situation known as degradation problem. He et al. (2016) introduced the Residual Networks (ResNets), by using a new approach to the problem of learning in deep neural networks to address this issue. Instead of training a network to learn a desired feature map H from an input map x (identity), it makes more sense to teach the model to learn $F = H - x$, the residual map that added to x, gives the desired output. This way, the optimization problem becomes easier.

Figure 2.5: Comparison between training of plain networks and ResNets on ImageNet

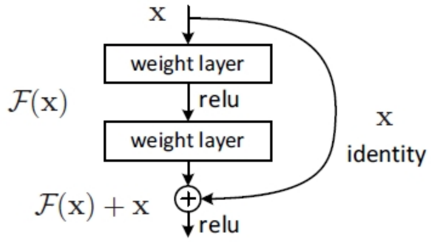


Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. The residual networks have no extra parameter compared to their plain counterparts.

Source: He et al. (2016).

To add this identity information from an early layer to another, they use skip connections, that bypass some layers and feed their output to others, as shown in Figure 2.6.

Figure 2.6: Residual learning: a building block


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Source: He et al. (2016).

He et al. (2016) states that, even when the net is "not overly deep" (18 layers for ResNet18) and the optimizer (they considered the Stochastic Gradient Descent) is still able to find good solution in plain nets, ResNet makes the optimization task easier by having a faster convergence at an early stage.

2.3.1.2 MobileNetV2

Sandler et al. (2018) introduced a neural network architecture designed for mobile and resource constrained environments (like embedded systems). It is based in depthwise separable convolutions, for which the base idea is to substitute a full convolutional operation by a drop-in replacement composed of two separate layers: a depthwise convolution, performing lightweight filtering by applying a single convolutional filter per input channel; and a pointwise convolution (1 × 1 convolution), that builds new features through the computation of linear combinations of the input channels.

2.3.2 Fully convolution network (FCN)

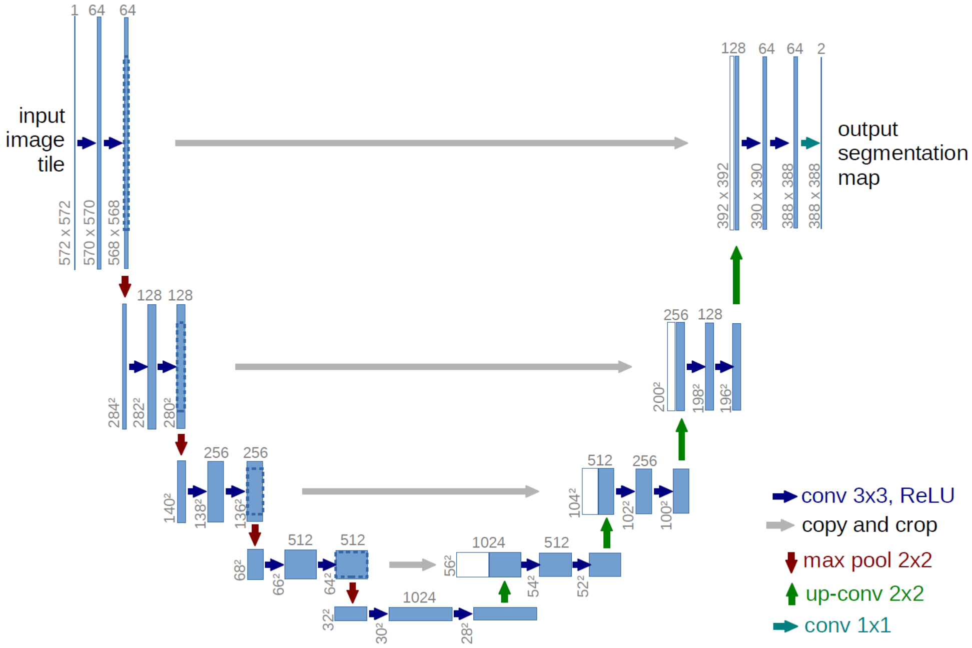
Semantic segmentation consist in classifying each pixel of an image to one or more classes. This can be done by a fully convolutional network, which is similar to a CNN, but it does not contain any "dense" (fully connected) layers. Instead, FCNs contain 1x1 convolutions to handle the classifier aspect performed by the fully connected layers. This change allows the network to perform semantic segmentation of images of different sizes and aspect ratios without using and combining parts of the input image. This is specially relevant if resizing the images can distort important features.

Another pertinent aspect is the use of downsampling and upsampling in FCN architectures. The first part of the model downsample the spacial resolution of the image, obtaining feature mappings with finer information at each convolution, to discriminate each class. The precise location is lost, but it can be recovered in an upsampling stage, that transforms the low resolution map back to the original resolution of the input image.

2.3.2.1 U-Net (UNet)

U-Net (RONNEBERGER; FISCHER; BROX, 2015) is a deep learning architecture for semantic segmentation based on a fully convolutional neural networks. It was originally designed for biomedical image segmentations, a field where typically there is few training data available (for example around 30 annotated images per application). Figure 2.7 shows U-Net architecture. It is basically consisted of two parts: a contraction path (down-sampling path, encoder) and an expansion path (decoder).

Figure 2.7: U-Net architecture (example for 32x32 pixels in the lowest resolution)



Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The *x*-*y*-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Source: Ronneberger, Fischer and Brox (2015).

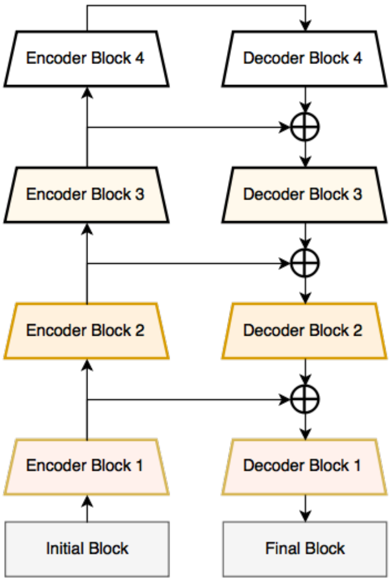
The encoder is composed of a sequence of convolutions and max pooling operations, as it is usual in convolution neural networks. This results in a spatial contraction where gradually the classes in the image are identified, but the information of their precise position decreases. A standard classification network would end at this point, having a classifier composed of a dense neural network connected on the top of the encoder block.

The decoder is a sequence of up-convolutions and concatenations (skip connections, the gray horizontal lines in the center of "U" in Figure 2.7) with high-resolution features from the down-sampling path. This way, the U-Net combines spatial information from the contraction path with the expansion path to retain good spatial information of the identified classes in the output segmentation map.

2.3.2.2 LinkNet

LinkNet (CHAURASIA; CULURCIELLO, 2017) is defined as a light deep neural network architecture designed for performing semantic segmentation. Figure 2.8 shows the LinkNet architecture. It also consists of an encoder and a decoder, with the information being shared by an addiction operation instead of a concatenation after each down-sampling block. The encoder used in Chaurasia and Culurciello (2017) was a ResNet18, and batch normalization between convolutional layers followed by ReLU is used.

Figure 2.8: LinkNet architecture



Source: Chaurasia and Culurciello (2017).

2.3.3 Transfer learning

Transfer learning consists of taking a model (or part of it) that has learned features for one task and applying it in another context. One can benefit not only from the architecture developed but also reuse the weights learned as a starting point or even "freeze" layers, which means their weights are not updated. For instance, a CNN model trained for classifying animals is likely to be useful when used in another task where the goal is to classify only a subset of these animals or even different species.

This is especially useful when there is not as much data available to train a complete model for the new task. In the case of CNNs with a small dataset (a few thousands images or less) and a similar task to the domain of the pre-trained model's dataset, a strategy is to freeze the group of convolutional layers and train only the classifier. However, in a different domain with a small dataset, part of the convolutional layers should be trained.

According to Chollet (2020), a typical transfer learning workflow is:

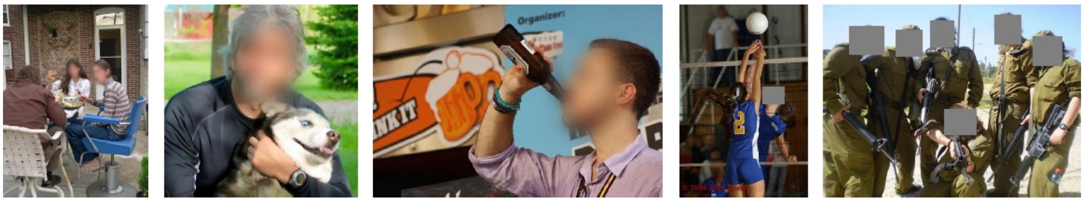
- take layers from a previously trained model, loading pretrained weights into it;
- "freeze" the layers, to avoid losing their capacity to recognize features;
- add trainable layers on top of the frozen ones;
- train the new model on the dataset of the current task.

Fine-tuning can then be performed, which may improve the performance of the model. In this case, a low learning rate is advisable, as the risk of overfitting is significant if large weight updates are applied (CHOLLET, 2020).

2.3.3.1 ImageNet

ImageNet is a large scale dataset containing over 14 million images that have been annotated by the ImageNet project (DENG et al., 2009) to indicate what objects are pictured. Even if ImageNet is not a dataset focused on people, faces or eyes, this dataset contains many such elements. There are 3 people categories in the 1000 categories of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (RUSSAKOVSKY et al., 2015). However, Yang et al. (2022a) annotated 1431093 images in ILSVRC, resulting in 562626 faces from 243198 images (17% of all images have at least one face). They also indicated that many non people categories have more than 90% images with faces co-occurring with the object of interest, as in Figure 2.9, posing a potential privacy threat.

Figure 2.9: Some non people categories in ImageNet Challenge



Example images (with faces blurred or overlaid) of barber chair, husky, beer bottle, volleyball and military uniform.

Source: Yang et al. (2022a).

2.3.4 Loss Function (Cost function)

A loss function, as known as cost function, is used to quantify the error between the output of an algorithm and the ground truth. During training, the goal of the optimizer is to minimize the loss function. There are several losses used in semantic segmentation, and they can also be combined, as each loss may prioritize different features in the mask. Typically, accuracy is not a good metric for semantic segmentation, particularly because of class imbalance. The inaccuracy of minority classes is hide by the accuracy of majority classes. This can be the case for a mask whose largest area is background.

2.3.4.1 Binary Crossentropy Loss

Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events. Cross-entropy loss, also called logistic regression loss or log loss in classification problems, is defined in Equation 2.3.

$$CE = -\sum_{i}^{C} t_i \log\left(\sigma\left(s_1\right)\right) \tag{2.3}$$

where

- C - number of classes;
- $t_i \in \{0,1\}$ - target, the ground truth value for each class i in C classes;
- s_i - score outputted by the model for each class i in C classes.

For binary classification problems (when $C = 2$), the Cross-entropy loss is called Binary Cross-entropy loss and can be defined as in Equation 2.4.

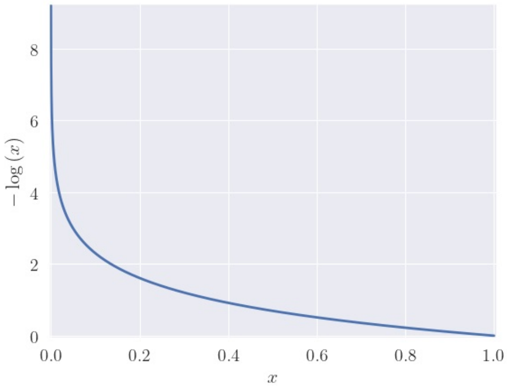
$$CE = -t_1 \log\left(\sigma\left(s_1\right)\right) - \left(1-t_1\right)\log\left(1-\sigma\left(s_1\right)\right) = \begin{cases} -\log\left(\sigma\left(s_1\right)\right), & \text{if } t_1 = 1 \\ -\log\left(1-\sigma\left(s_1\right)\right), & \text{if } t_1 = 0 \end{cases} \tag{2.4}$$

where

- $t_1 \in \{0,1\}$ - target, the ground truth value for the class 1;
- s_1 - score outputted by the model for the class 1.

As shown in Figure 2.10, the negative logarithm function penalizes predictions, giving a high loss value for the worst predictions. In this case, $s_2 = 1 - s_1$ and $t_2 = 1 - t_1$.

Figure 2.10: Negative logarithm function



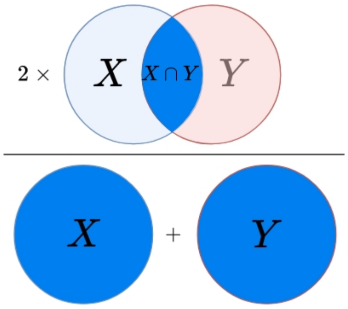
.

Source: Author.

2.3.4.2 Dice

Dice coefficient score is two times the area overlap between the prediction segmentation mask and the original segmentation mask (the ground truth) divided by the area of the prediction segmentation mask added by the area of the original segmentation mask. The Figure 2.11 illustrates the concept of the Dice Coefficient.

Figure 2.11: Dice Coefficient



Source: Author.

The Dice coefficient can be expressed by the Equation 2.5.

$$\text{Dice score} = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} \tag{2.5}$$

where

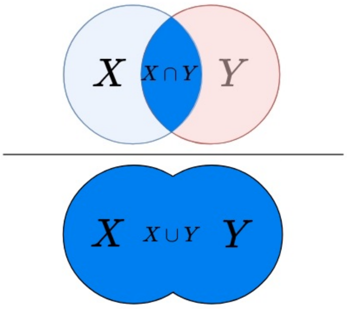
- X is the predicted set of pixels;
- Y is the ground truth.

Milletari, Navab and Ahmadi (2016) proposed using an objective function based on Dice coefficient maximization as loss, as an alternative to using weights to increase the importance of foreground compared to the rest of the volume (background).

2.3.4.3 Intersection over Union (IOU)

Intersection over Union is the area overlap between the prediction segmentation mask and the original segmentation mask (the ground truth) divided by the area of union between the prediction segmentation mask and the original segmentation mask. The Figure 2.12 illustrates this concept. Values closed to 1 are ideal.

Figure 2.12: Intersection over Union



Source: Author.

The Intersection over Union can be expressed by the Equation 2.6.

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} \tag{2.6}$$

where

- X is the predicted set of pixels;
- Y is the ground truth.

IoU is a metric featuring a function not differentiable, not suitable for being used as a loss function for training. IoU is correlated to the Dice coefficient.

2.4 Overfiting

When a machine or deep learning model adapts so well to the training data, but not for data not used for training, an overfitting has occurred. Overfitting usually happens when there is not enough representative training data to a given problem or the complexity of the model is too large.

Two forms of handling overfitting are to increase the number of samples of the training dataset (provided that the data is pertinent to the problem) and to reduce the complexity of the model. In the first case, the samples added have to be meaningful to this condition. If the original dataset is imbalanced, care should be taken to not further aggravate the problem. The idea behind the second case is that a simpler model will have to store less features, so, if well-trained, it will likely focus on the most relevant features and has a better chance to generalize better.

2.4.1 Regularization

Regularization is used to modulate the entropy of the model, forcing the weights to smaller values, and thus reducing the model complexity and forcing it to learn the most relevant feature of data. A common type is the L2 regularization, expressed as the sum of the squares of all the feature weights. The optimizer then searches to minimize the total loss, as in Equation 2.7.

$$\text{Total Loss} = \text{Loss} + \lambda \sum_{j=1}^{M} {\omega_j}^2, \tag{2.7}$$

where:

- $\lambda \in [0,1]$ is the regularization rate, the hyperparameter that controls the penalty;
- Loss is the cost function computeted;
- ω_j corresponds to the synapses (weights) of the ANN.
- M is the total number of network parameters, i.e., the number of synapses.

Typical values of λ for convolutional deep learning models are small. In their deep convolutional model trained in ILSVRC - 2010 subset of ImageNet (roughly 1.2 million training images, 50000 validation images, and 150000 testing images), Krizhevsky, Sutskever and Hinton (2012) commented that a small weight decay of 0.0005 had not only a regularize effect, but also, helped to reduce the model training error. Simonyan and Zisserman (2014) also used a weight decay (L2 penalty multiplier) of 0.0005, whereas Chollet (2017) informs that Inception V3 model used $4 \cdot 10^{-5}$, that have been carefully tuned for performance in ImageNet. This value was judge suboptimal for the Xception model where it was instead settled for $1 \cdot 10^{-5}$, although an extensive search for the optimal weight decay rate was not performed. He et al. (2016) used 0.0001 for the ResNet models for the ILSVRC-2012.

Google for Developers (2022) points out that the learning rate and the regularization rate are close related, in that a high L2 regularization tend to drive feature weights closer to 0, while a lower learning rate (together with early stopping technique) often have this effect, because the steps away from 0 aren't as large. Consequently, tweaking learning rate and lambda simultaneously may have confounding effects.

2.4.2 Early stopping

Early stopping is a technique that stops training before it fully converges (for example, before training loss finishes decreasing or when a monitored metric does not improve anymore). As training a model for too many epochs can lead to overfitting, early stopping can prevent these decrease in performance. In practice, often there is an implicit early stopping when training a model with a limited number of epochs. Saving the model (or its weights) that achieved the best performance so far for some metric or loss can also be considered a form of early stopping and regularization.

2.4.3 Data augmentation

Data augmentation is a technique that applies minor changes to the data and can be used to artificially extends the diversity of the training dataset. This helps to avoid overfitting to the training dataset, because it adds some variability to the data, that can be transformed differently on every epoch. Geometric and color space transformations operations, like flipping, resizing, cropping, changing brightness and changing contrast are examples of modifications used when augmenting images.

In tasks like semantic segmentation, it is important to match up images and masks when performing operations like flipping, zooming and rotating. Likewise, the transformations should match changes that the model can face once training is done. Data augmentation can also improve the performance of the models, provided that the changes are meaningful, not "too aggressive" and the original dataset is not "too small" (which depends on task and data available). For example, when determining the state of the eye, the model is unlikely to have a face upside down as an input.

Architectures like U-Net are used in biomedical image processing for image segmentation, where thousands of images are not available. Ronneberger, Fischer and Brox (2015) utilized U-Net to the segmentation of neuronal structures in electron microscopic recordings in a dataset of 30 images (512x512 pixels) and extensive data augmentation to the available training images.

Zheng et al. (2022a) utilized U-Net and 1019 pairs of images and manual annotations of right eyes obtained with a Keratograph 5M (512x512 pixels) and performing data augmentation on-the-fly (random flipping along the vertical axes, translation by -10% to 10% per axis, rotation from -20 to 20 in degrees, and scaling from 0.9 to 1.1.)

It should be noted that the augmented images are still highly correlated. Because of this, issues like data imbalance by absence of a class are not addressed by common transformations like flipping and translations of data augmentation.

2.4.4 Batch normalization

Batch normalization is a technique introduced by Ioffe and Szegedy (2015) to accelerate deep network training, improving generalization and convergence. It is introduced between the layers of the models, performing a transformation that maintains the mean output of the layer close to 0 and the standard deviation close to 1 for each batch. Equation 2.8 represents this technique.

$$y = \gamma \frac{x-\mu}{\sigma} + \beta \tag{2.8}$$

where:

- y is the output (scaled and shifted value representing the batch normalization result);
- γ is the scale (parameter to be learned);
- x is the input;
- μ is the mean of x in mini-batch;
- σ is the standard of x in mini-batch.
- β is the shift (parameter to be learned).

It should be noted that the size of the batches should not be too small, otherwise the statistics of the batch will not represent the actual dataset.

2.5 Hyperparameter tuning

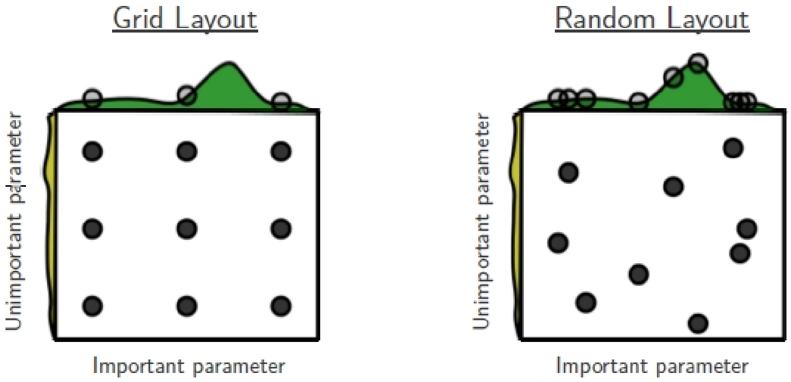
In machine learning and deep learning, the parameters that are set before the start of the training are called hyperparameters. These parameters can define the topology of a neural network (like its depth or the number of neurons per layer), the learning rate used in a gradient descent type of algorithm, the regularization applied to each layer, and many other characteristics of the model and of the learning process.

When using transfer learning with deep learning, many parameters regarding the topology of the network are already set. However, hyperparameters related to the learning processes, which are key to avoiding overfit, must be set. Manually setting the parameters and analyzing the learning curve is useful for verifying if the model is adapted to the data and judging its performance and generalization. While this is useful, searching the hyperparameter space may be relevant to obtaining the best score.

Two strategies to perform this search automatically are Grid Search and Random Search. Grid Search tests all possible combinations of values from the subset of hyperparameters chosen, while Random Search consists of independent draws generating trial sets of hyperparameters from a uniform density in the same configuration space as would be spanned by grid search.

Bergstra and Bengio (2012) suggests that there may be a small reduction in efficiency in low-dimensional spaces using random trials, but a large improvement in efficiency in high-dimensional search spaces can be achieved. While Random Search does not guarantee to find the best score in the sample space as Grid Search does, it is likely to find a good combination faster as it allows exploring the choices of hyperparameters more widely than Grid Search. This is especially relevant if some hyperparameters are more relevant than others, as depicted in Figure 2.13.

Figure 2.13: Grid and Random Search of nine trials



Grid and Random Search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g.

Source: Bergstra and Bengio (2012).

2.6 Image post-processing techniques

Connected-component analysis (CCA), also known as connected-component labeling and blob extraction, is a traditional image post-processing algorithm. It's an application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. It allows filtering the noise in a segmentation mask and to extract the most relevant components from it, as well as statistics, like the area of components and their centroids.

For the 4-way connectivity, only the pixels that share at least one border are considered connected. In the 8-way connectivity, pixels that are in the diagonals, i.e., share at least a vertex, are also considered. Connected-components in a binary mask can have their contours analyzed by contour detection. This can be done by the comparison of each pixel with its neighbors and extracting the location of pixels where there is a color change.

3 RELATED WORK

3.1 Blink rate level, blink inducing and blink detection

Lee et al. (2021) have applied a sensor-embedded chair to obtain sitting postural behavior data that was compared to eye blinking data collected with Dikablis head-mounted eye tracking system (Ergoneers). A blink was defined as the pupil not being detected in the eye-tracking system for more than 100 ms. Compared to a high eye blink rate condition, low eye blink rate condition was related to less overall postural variability and greater extent of forward bending posture.

There are other systems based in external hardware, like an electrooculography based blink detection to prevent CVS by Pal et al. (2014), eyewear systems for helping users follow the $20-20-20$ rule by monitoring user's screen viewing activities (MIN et al., 2019) or tracking blinks by means of infrared reflections from the wearer's cornea or eyelid (DEMENTYEV; HOLZ, 2017). Most of the related work analyzed here will focus in systems that do not depend on external hardware and uses a webcam, that is commonly available for computers and built-in for some models of VDT (laptops and smartphones).

3.1.1 Blink Animation Software to Improve Blinking and Dry Eye Symptoms

Nosch et al. (2015) introduce "Blink Blink" software based on an animation to increase blink rate and reduce dry eye syndrome symptoms during daily computer use. Two semi-transparent bars move from the top and bottom of the screen towards the center of the screen, independently of the application in use on the computer. The percentage of the screen covered, the opacity of the bars, the duration of the animation and the appearance interval can be individually configured to best suit the user. In the study, each animation was set to 600 ms, with 20% coverage for each bar and 25% opacity. The aim of these settings was that the animation would be noticeable, but the user's concentration would not be substantially affected, nor the use of the mouse and keyboard during the presentation.

3.1.2 Stimulating a Blink: Reduction of Eye Fatigue With Visual Stimulus

Crnovrsanin, Wang and Ma (2014) investigated four different types of eye-blink stimulus: screen blurring, screen flashing, border flashing, and pop-up notifications to increase blink rate of 13 computer users, which rated each stimulus type in terms of effectiveness, intrusiveness, and satisfaction in an active image task (spotting the difference between images). Results from the studies showed that all stimuli are effective in increasing user blink rate with screen blurring being the best. The Blur stimuli causes the screen to slowly blur until the user blinks.

Crnovrsanin, Wang and Ma (2014) comments that user frustration and interruption of workflow play a big role in individuals adopting these systems, pointing out that correct blink detection plays a huge role in how intrusive the subject feels a stimulus method is.

3.1.3 RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environment

Cortacero, Fischer and Demiris (2019) addresses the estimation of blinks and gaze direction together and introduces the RT-BENE dataset. This dataset has more than $200,000$ eye images, with more than $10,000$ with eyes closed. The technique of *oversampling* with weights is used to deal with the class imbalance. The authors make use of CNN (Mask R-CNN) and transfer learning (MobileNetV2, ResNet50 and DenseNet121).

Cortacero, Fischer and Demiris (2019) also points out that CNNs have been used to partly overcome the limitations of traditional methods based on feature extraction such as *scale-invariant feature transform* (SIFT) and *Histogram of Oriented Gradients* (HOG) followed by a classification stage. According to the authors, the accuracy of these methods is reduced for head positions with extreme angles and for varying skin tones and lighting. CNNs also allow the estimation of blinks for faces in a non-full frontal position. This is especially interesting when considering the use of a second monitor (second screen) by the user.

An alternative to the use of CNNs with traditional methods is to consider a sequence of images, instead of each image isolated. Fogelton and Benesova (2016) considered a state machine, while Fogelton and Benesova (2018) use a recurrent neural network.

3.1.4 Eyeblink, de Andrej Fogelton

Andrej Fogelton, author of research in the area of blink detection and analysis, has developed the Eyeblink software, available at <https://www.blinkingmatters.com>. This software alerts the user according to the time of use and number of low blinks, in order to prevent CVS and Dry Eye Syndrome. Blink completeness is not analyzed.

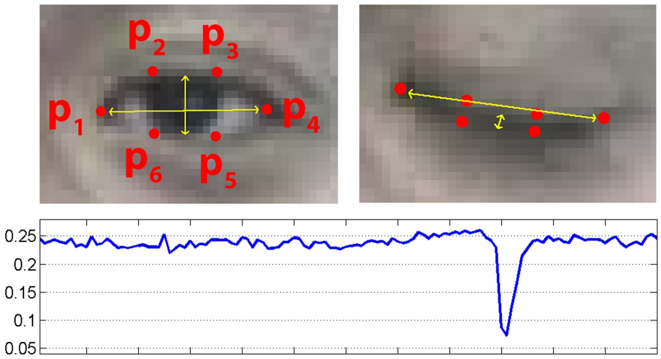
3.1.5 Eye Aspect Ratio (EAR)

Soukupová and Cech (2016b) proposed to distinguish between open and closed eyes using 6 landmarks points and analyzing the Eye Aspect Ratio (EAR). Equation 3.1 shows how EAR is computed.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\ \|p_1 - p_4\|} \tag{3.1}$$

where $p_1, \cdots, p_6$ are 2D landmark locations, depicted in Figure 3.1.

Figure 3.1: EAR: Open and closed eyes with landmarks p_i automatically detected



The eye aspect ratio EAR in Equation 3.1 plotted for several frames of a video sequence. A single blink is present.

Source: Soukupová and Cech (2016b).

Some EAR properties are:

- it has an almost constant value when the eye is open, and the value decreases and then increases again during a blink;
- its value is close to 0 while the eye is closed;
- it is partially person and head pose insensitive;

- it is theoretically invariant to uniform image scaling and in-plane face rotation.

In a simplified approach, if the EAR is greater than the threshold of 0.20, then the system identifies the region of interest as opened eyes. If the EAR is less than this threshold, then it is classified as a closed eye.

Low values of EAR may indicate that the subject is performing a facial expression (person squinting eyes, screaming, smiling, disgusted), closing the eyes for longer periods than a blink, yawing or even that EAR is reproducing a short fluctuation of the landmarks.

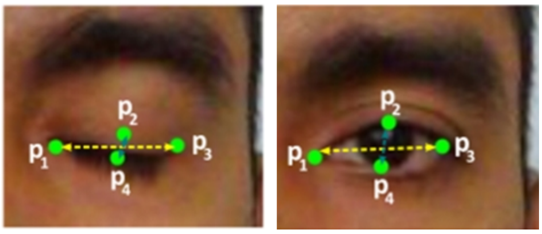
Soukupová and Cech (2016b) also experimented using a Support Vector Machine (SVM) classifier that takes a temporal window of ±6 frames (for 30 FPS videos tested) into account to classify the blinks. SVM is a machine learning technique that can separate classes of data by generating a hyperplane that tries to maximize the separation between the classes of elements. Since eye blinking is performed by both eyes synchronously, an average of the EAR of both eyes is computed.

Yin et al. (2022) used the EAR with a threshold of 0.20 to detect blink frames. After that, the authors used image enhancement strategies and magnification before calculating the vertical distance between the upper eyelid and eye corner to finally recognize incomplete blink.

3.1.6 Eye Aspect Ratio based variations

Huda, Tolle and Utaminingrum (2020) proposed a modified EAR, using 4 points around the eye, as shown in Figure 3.2. According to the authors, the 4 greatly affect the measurement of closed eyes, and this reduction has a significant impact on the calculation process and improves computational time to work in real-time on a mobile device.

Figure 3.2: 4 Points EAR



(a) Closed Eye. (b) Open Eye.

p_1 denotes the point at the eye gland, p_3 denotes the point at the other corner of the eye, p_2 denotes the top point at upper lid and p_4 denotes the point at the lower lid.

Source: Huda, Tolle and Utaminingrum (2020).

Equation 3.2 demonstrates an estimation of 4 points in the eye area.

$$EAR_{4points} = \frac{|p_2 - p_4|}{|p_1 - p_3|} \tag{3.2}$$

where

- p_1 - point at eye gland;
- p_2 - top point at upper lid;
- p_3 - point at the other corner of the eye, opposed to p_1;
- p_4 - point at lower lid.

Table 3.1 shows the result of closed eye identification with EAR (4 points) threshold variation with 10 people of four different types of Indonesian races (HUDA; TOLLE; UTAMININGRUM, 2020). Test have been done with 640 x 480 resolution with 8 FPS acquisition by an Asus Zenfone 2 ZE551M smartphone camera. Huda, Tolle and Utaminingrum (2020) have considered 0.24 the optimal threshold value for detecting closed eyes.

Table 3.1: Result of closed eye identification with variation of EAR (4 points) threshold

$N°$	**Mean EAR of Opened Eyes**	**Number of Closed Eyes**	**Result of System Detection with variation of EAR (4 points) threshold**				
			0.20	**0.22**	**0.24**	**0.26**	**0.28**
1	0.26	20	17	19	18	3	0
2	0.26	20	16	18	18	5	0
3	0.27	20	13	18	19	8	3
4	0.29	20	17	18	18	17	1
5	0.29	20	17	18	19	18	10
6	0.29	20	18	18	19	18	13
7	0.31	20	0	10	17	18	18
8	0.31	20	10	16	18	18	17
9	0.32	20	10	12	17	18	18
10	0.33	20	13	16	18	19	19
Mean			13.1	16.3	18.1	14.2	9.9
Percent			65.5%	81.5%	90.5%	71.0%	49.5%

Source: Huda, Tolle and Utaminingrum (2020).

As Soukupová and Cech (2016a) experiments have shown, the facial landmarks are very precise even at low resolution, allowing a SVM blink detector based on EAR to give good results even at interocular distance of only 10 pixels. But EAR is affected by the number of pixels in the input image. According to Kuwahara et al. (2022), when the face is small, the noise in the EAR becomes large, which is one of the causes of the decrease in blink detection accuracy. This has being also studied by Ye et al. (2022), as shown in Appendix A EAR and DLib model performance in images as function of image quality. One interesting aspect is that the model positioning the landmarks in Ye et al. (2022) seems to generally perform better with open eyes images.

Kuwahara et al. (2022) have used blink detection based on the Eye Aspect Ratio Mapping (EARM), proposed in Kuwahara et al. (2021a), to estimate eye fatigue. Equation 3.3 shows the definition of EARM.

$$\begin{aligned} EARM(t) = EAR\left(t-\frac{X+1}{2}\right) \,+\, EAR\left(t-\frac{X-1}{2}\right) \,+\, EAR\left(t+\frac{X-1}{2}\right) \\ +\, EAR\left(t+\frac{X+1}{2}\right) \,-\, 4\times EAR(t) \end{aligned} \tag{3.3}$$

where t is the number of frames since the start and X is an odd number of frames per eye blink. In Kuwahara et al. (2021b) and Kuwahara et al. (2022), this number is defined as 9 and 11, respectively.

Kuwahara et al. (2021b) test with 5 subjects have found that both EAR (with fixed threshold of 0.20) and EARM may benefit of normalizing facial image, with EARM with normalized face displaying the best results. However, in some cases, normalizing was not beneficial. The authors believed that this was possibly because part of the face was occluded, which inhibited an improvement in the recognition rate of the face.

Maior et al. (2020), constructing a drowsiness detection model using EAR, has tested two methods to differentiate short and long blinks. The first used a calibration procedure comparing a subject's neutral face (when reading) to a smiling face in order to define the EAR threshold. They reported that this method leads to many false positive warnings from trivial expressions (e.g. talking). The second method was the concatenation of 15 consecutive EARs values from 13 users, classified in open eye, short blink, and long blink, to train different machine learning models. The model selects inputs every 5 frames and a blink was considered if the touch of eyelids occurred in the 5 central frames. A SVM model was chosen, and user-specific data was aggregated with existing training data to train a new SVM model, and the updated model is used for state detection. Specifically, a pre-trained model automatically classifies user's new data during runtime and open eyes data sequences not near to blinks are added to the training data. This SVM-based model with personal feedback have an improved accuracy, which highlights the value of user specific calibration.

Dewi et al. (2022) have proposed a modified eye aspect ratio (Modified EAR), that is a threshold to determinate eye status (open or closed) and correspond to the average of the result of Equation 3.4 (EAR for closed Eyes) and Equation 3.5 (EAR for open Eyes). The entire video was analyzed in this case.

$$EAR_{Closed} = \frac{\|p_2 - p_6\|_{min} + \|p_3 - p_5\|_{min}}{2\ \|p_1 - p_4\|_{max}} \tag{3.4}$$

$$EAR_{Open} = \frac{\|p_2 - p_6\|_{max} + \|p_3 - p_5\|_{max}}{2\ \|p_1 - p_4\|_{min}} \tag{3.5}$$

3.2 Analysis of complete and incomplete blinks

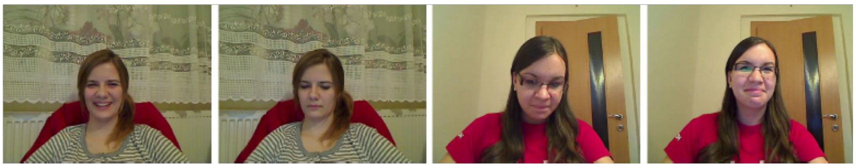
3.2.1 Eye blink completeness detection

Fogelton and Benesova (2018) use a larger and particularly challenging dataset for computer vision tasks, given inadequate lighting conditions: Researcher's night re-annotated. 100 individuals with 1849 annotated blinks with hectic and disorganized background. People are often acting naturally, wearing glasses (about 20%), touching their faces, moving their heads or talking with someone (FOGELTON; BENESOVA, 2016).

On the website https://www.blinkingmatters.com/, maintained by Andrej Fogelton, the annotations of the blinks from the datasets ZJU, Talking Face, Eyeblink8, Silesian5 (on demand for the videos) and Reseacher's night (on demand) are available. They all were analyzed for blink completeness.

Introduced by Drutarovsky and Fogelton (2015) and then re-annotated, Eyeblink8 dataset consists of 8 videos with 3 individuals (1 wearing or not wearing glasses) with their faces directed towards the camera most of the time and more than 800 blinks with each eye evaluated separately (762 completes and 44 incompletes). This dataset has the advantage of being available directly to the community, and it is specially interesting for investigating eye blink completeness in the context of CSV, as it represents similar conditions to the ones a computer would be used, and the 640 x 480 resolution videos were recorded at 30 FPS, a frame rate that is adequate for eye blink completeness analysis (ZHENG et al., 2022a). Figure 3.3 shows some sample snapshots from Eyeblink8 dataset.

Figure 3.3: Sample snapshots from Eyeblink8 dataset



Source: Drutarovsky and Fogelton (2015).

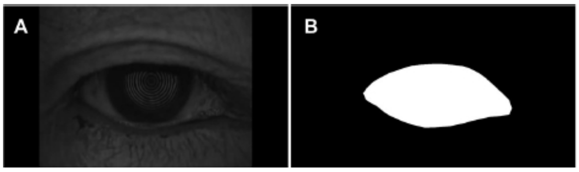
In Fogelton and Benesova (2018), they make use of a recurrent neural network. This type of ANN has internal state memories to process sequences of data and is applied by the authors to estimate blink completeness based on optical flow for motion detection.

3.2.2 Impact of Incomplete Blinking Analyzed Using a Deep Learning Model With the Keratograph 5M in Dry Eye Disease

Zheng et al. (2022a) analyzed the impact of incomplete blinking on dry eye syndrome using a deep learning model with images obtained from a Keratograph 5M. This instrument is a corneal topographer; it has a high-resolution camera and allows the examination of the meibomian glands, non-invasive examination of the tear film break-up time, measurement of the height of the tear meniscus, and evaluation of the lipid layer.

Figure 3.4 shows the original and labeled image after manual processing. In part B, the image was manually filled in the interpalpebral region with white color and with black color in the rest. The interpalpebral zone was annotated by a single investigator.

Figure 3.4: Original and labeled image after manual processing



The concentric circular pattern is generated by the Keratograph 5M.

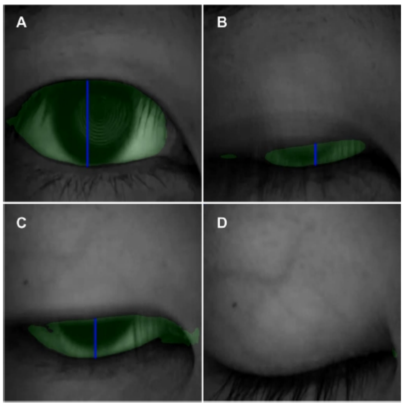
Source: Zheng et al. (2022a).

The original image and its manual annotation were resized to 512 by 512 pixels using the nearest neighbor interpolation feature of *Python Image Lybrary* (version 6.2.0) and then merged as a single set. A total of 1019 images were used, collected and randomly distributed into three distinct training, validation and test sets in a ratio of 8 : 1 : 1. These images were used to train a U-Net convolutional network to segment the exposed palpebral fissure. The model was implemented in the Python programming language (version

3.7.4) with the Keras library (version 2.1.6).

Once the model is trained, the average relative interpalpebral height (IPH) is used to determine whether a blink is complete or not. First, the maximum value of the IPH was determined, which is done by analyzing the entire video. When a certain blink occurs, if it is greater than 30% of the maximum IPH value, this blink is considered incomplete in the scope of this work. Figure 3.5 shows the detection of one blink.

Figure 3.5: Detection of a blink for Keratograph 5M images



The interpalpebral zone is colored green and the average relative interpalpebral height (IPH) is represented by the blue segment.

Source: Zheng et al. (2022a).

3.2.3 Evaluation of VDT-Induced Visual Fatigue by Automatic Detection of Blink Features

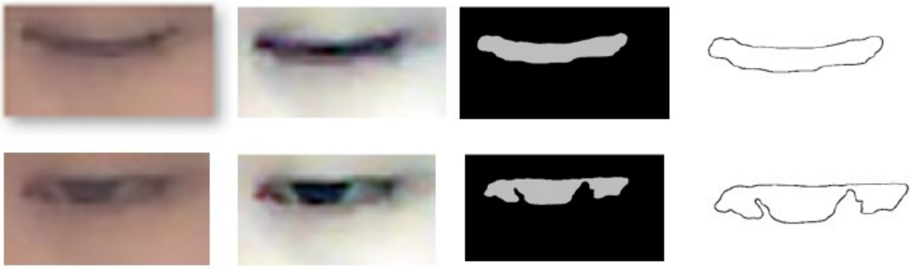
This study evaluates the progression of visual fatigue induced by the use of VDT. Yin et al. (2022) detect blinks and incomplete blinks through automatic video detection (frame rate of 60 FPS) using computer vision techniques.

The algorithm used consists of:

- reading the image from the recorded video;
- downsample for 600 by 450 of the faces images to reduce computation;
- applying advanced facial landmarks (set of regression trees to estimate the position of landmarks on the face from a subset of pixel intensities; real-time performance with high quality prediction) to detect the face and localize the eyes;
- obtaining a frame containing blinking eyes, based on the variation of the EAR determined from the markers positioned in the previous step;

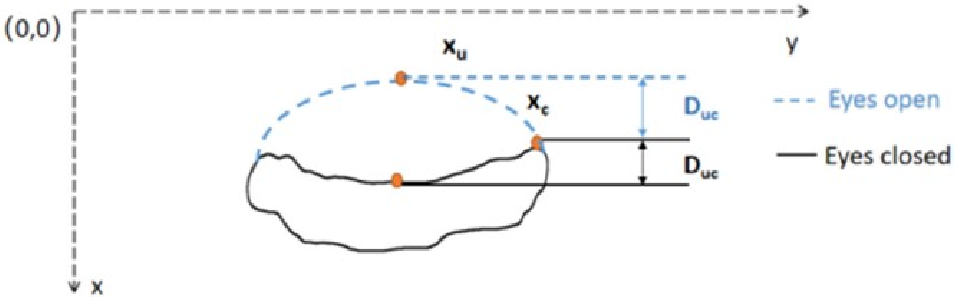
- application of *single scale retinex* to improve brightness, contrast and sharpness of a grayscale image through a combination of spatial and spectral transformations, with the eye image being enlarged and subsequently converted to grayscale and binarized, and then applying adaptive threshold segmentation (see Figure 3.6);
- calculation of the distance between the upper eyelid and the corner of the eye ($D_{uc} = x_u - x_c$), as outlined in Figure 3.7 ;
- recognition of incomplete blinks (with threshold determined per individual);
- extraction of blink features.
- saves the features in a file.

Figure 3.6: Contour extraction process



Source: Yin et al. (2022).

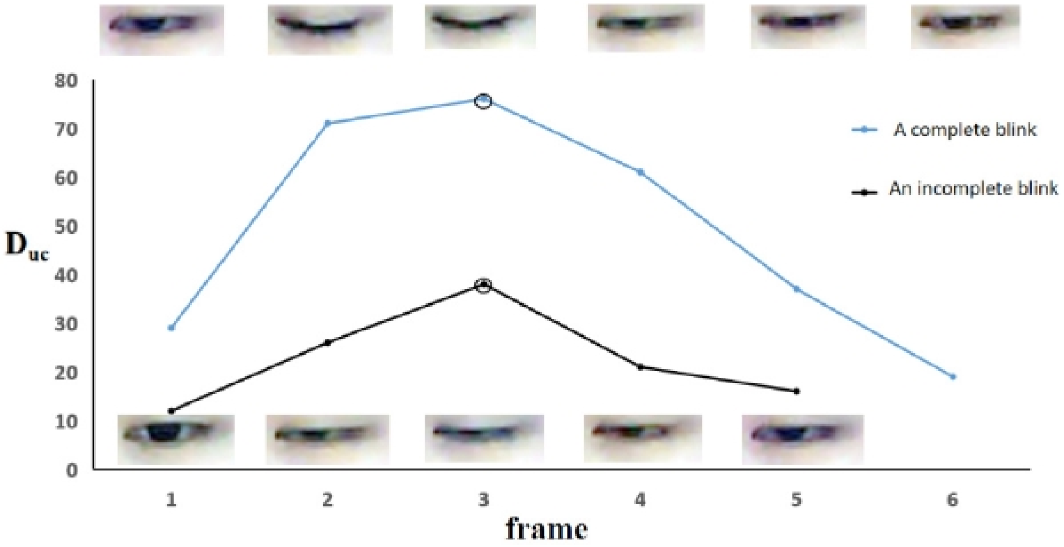
Figure 3.7: Vertical distance between the upper eyelid's midpoint and the eye's corner



Source: Yin et al. (2022).

The Figure 3.8 shows the change in D_{uc} during one blink.

Figure 3.8: Change in D_{uc} during one blink



Source: Yin et al. (2022).

Manual labeling of complete and incomplete blinks was performed for all individuals (about 3000 blinks, including 500 incomplete). D_{uc} was calculated for each frame with blinking images and $maxD_{uc}$, the maximum D_{uc} obtained for each blink, was determined. The median value of $maxD_{uc}$ for each individual in the 10 min was computed (mmD_{uc}). If the $maxD_{uc}$ was less than 75% of the mmD_{uc}, the blink was considered incomplete; otherwise, complete. The subject-specific threshold was used in the following 110 minutes of recorded videos to detect complete and incomplete blinks.

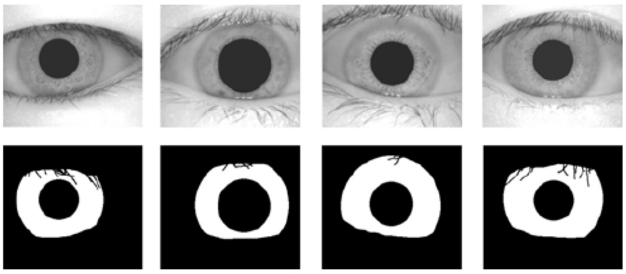
3.3 Perorbital and palpebral fissure segmentation

3.3.1 End-to-end Iris Segmentation Using U-Net

Lozej et al. (2018) inform that traditional iris segmentation techniques have typically been focused on hand-crafted procedures. More recently, researchers are increasingly looking towards CNNs to further improve on the accuracy of existing iris segmentation techniques. The authors then present an iris segmentation approach based on the popular U-Net architecture (RONNEBERGER; FISCHER; BROX, 2015), trainable end-to-end and, hence, avoiding the need for hand designing the segmentation procedure.

The CNN-based segmentation model proved to be successful at segmenting the iris images, even with a reduced dataset (200 samples) and no data augmentation, outperforming four established techniques from the literature. The accuracy of the best model was reported to be $97,79\%$. The authors also reported the average intersection over union at the threshold with the best ratio between precision and recall: 0.912 ± 0.031. Figure 3.9 show some samples of the training data, while Table 3.2 shows the time and space complexity for U-Net models of different depths.

Figure 3.9: Illustration of the training data for iris segmentation



The top row shows sample images from the CASIA dataset (Chinese Academy of Sciences Institute of Automation (CASIA), 2003), the bottom row shows the annotated (binary) segmentation ground truth.

Source: Lozej et al. (2018).

Table 3.2: Time and space complexity for U-Net models for iris segmentation

Depth	Time/image	Maximum memory used
3	45 ms/image	5173 MB
4	60 ms/image	5181 MB
5	102 ms/image	5185 MB

A desktop was used with an Intel I7-2600k processor with 8 GB of RAM and a Nvidia GTX-1060 6 GB GPU. The grayscale images containing one eye were of size 320 × 320.

Source: Lozej et al. (2018).

3.3.2 PeriorbitAI: Artificial intelligence automation of eyelid and periorbital measurements

Brummen et al. (2021) describe an open source, fully automated AI system for segmentation and quantification of eyelid and periorbital measurements. They have used an UNet-style architecture with ResNet50 (ResNet with 50 layers) backbone with the last layer been a pyramidal pooling layer. Introduced in Zhao et al. (2017), this layer gets to model greater global context of the image, been specially useful for complex-scene images, where there are many classes to segment. Adam optimization algorithm

(KINGMA; BA, 2017) is used, with a starting learning rate of 0.001 decreased by a factor of 10 if the validation set average did not improve for 10 epochs. The batch size was set to 4.

418 photographs with image resolution of 6000x4000 pixels were taken. Approximately 80% and 20% of 397 images for training and validation (used for hyperparameter tuning), respectively, were segmented by two trained human graders. The remaining 21 images were used for a retrospective test set and were segmented by 3 human graders to evaluate the model. The images were splinted in half and resized to 256×256 pixels before input to the deep learning model. A prospective study was also performed latter.

Authors have used the dice coefficient to evaluate the accuracy of the model and human-generated segmentation using on expert as reference. The model obtained a Dice coefficient of 0.90 for the palpebral fissure.

The system performed within the range of inter-human variability with high precision despite multiple photographers (certified ophthalmic technicians), and a broader spectrum of conditions was considered in the data, making the model measurement spectrum more inclusive and more generalizable.

It should be noted that the images are of high resolution for the periorbital region, so eyes pixels occupy represent more image space.

3.4 Webcam and frame rate acquisition

Considering the discussion in 2.1.1 Eyeblink, an average blink takes between 150 to 300 ms (5 to 10 frames while 30 frames per second), making it feasible to be detected by a standard webcam, as the ones typically available in notebooks. Zheng et al. (2022a) demonstrated experimentally that an acquisition frame rate of 30 FPS provides more accurate and sensitive information than those of 8 FPS, as it is possible that some complete blinks are misjudged as incomplete in recordings with 8 frames per second. Yin et al. (2022) have used a frame rate of 60 FPS, while Fogelton and Benesova (2018) considered 30 FPS sufficient for blink completeness detection.

With the development of more powerful hardware and memory in these devices, and the increase of remote work, it is likely that better cameras will be available for notebooks and computers, specially in terms of resolution, as it is also a trend in mobile devices.

Blink analysis can be performed in a video or in real-time. Both situations are interesting. The first can be used in case of diagnostic, as in Zheng et al. (2022b) (with a Keratograph 5m) and Yin et al. (2022) (Spedal MF934H webcam), in the case of incomplete blink; and the second for prevention, as is the case of the software Eyeblink (FOGELTON, 2018).

3.4.1 Real-time use of blink analysis systems

The overall processing time including blink completeness algorithm should take less than $33,33$ ms for real-time use with a webcam recording at 30 FPS. Without considering a pipeline system, in which an image is being processed in parallel to the image acquisition (with a delay of one frame), the blink completeness step has to be performed in even less time, as face detection must be performed beforehand.

Table 3.3 shows an evaluation of processing time for Fogelton and Benesova (2016) blink detection algorithm (blink completeness was not considered). As it can be seen, the amount of time taken to perform face detection using OpenCV implementation of Viola–Jones Viola and Jones (2004) algorithm and also localize eye corners annotation automatically is quite fast, lesser than 12 ms for *Eyeblink8* dataset, for a computer with Intel core i5 3.3 GHz with CPU usage from 25 to 50% in 2016.

Table 3.3: Evaluation of processing time for Fogelton and Benesova (2016) blink detection algorithm

Dataset	Resolution	Viola–Jones + CLandmark	Blink detection
Talking face	720 x 576	10.3 ms	8 ms
Basler5	640 × 480	9.4 ms	10.6 ms
ZJU	320 x 240	6.5 ms	2 ms
Eyeblink8	320 x 240	11.6 ms	2.6 ms

The amount of time taken to localize eye corners annotation automatically and how much it takes to detect eye blinks is shown. In *Basler5* subject's face is very close to the camera. Times are measured on Intel core i5 3.3 GHz with CPU usage from 25 to 50%. Overall processing time (face, facial landmark and blink detection) is under 20 ms per image, which suits this method for real-time use.

Source: Fogelton and Benesova (2016).

Soukupová and Cech (2016a) indicate that the average time for processing one frame of Eyeblink8 dataset using EAR SVM is 19.2 ms. This time already includes the processing time taken for finding facial landmarks, so the algorithm can run in about $50-60$ FPS while using an ordinary laptop (64-bit Windows 8, Intel Core i7-5500U @ 2.4 GHz, 8 GB RAM).

3.5 Discussion on Related Work

There is a considerable number of studies regarding blink detection, with some focusing in the context of ophthalmological and ergonomics of the computer use. Nonetheless, not many studies have focused on detecting incomplete blinks and blink completeness, which remains a complex problem.

Deep learning approaches have been used to measure blink completeness. Fogelton and Benesova (2018) RNN best reported F1-score (a performance metric) for incomplete blinks was 0.49, while they achieve 0.758 for complete blinks, indicating that there is still space for improvement. His movement based method can be trapped by head movements. Appearance based methods can be trapped by make-up and eyelashes extensions. It is hypothesized here that, if enough quality data is provided, a fully convolutional network based architecture could overcome these challenges and provide reliable palpebral fissure segmentations that could be used to detect incomplete blinks by means of another algorithm, even with images where the eyes cover only a small region of the frames.

The use of CNNs could also be more flexible compared to Viola-Jones face detection algorithms that required frontal faces. Cortacero, Fischer and Demiris (2019) also points out that CNNs have been used to partly overcome the limitations of traditional methods based on feature extraction such as *scale-invariant feature transform* (SIFT) and *Histogram of Oriented Gradients* (HOG) followed by a classification stage. According to the authors, the accuracy of these methods is reduced for head positions with extreme angles and for varying skin tones and lighting. CNNs also allow the estimation of blinks for faces in a non-full frontal position. This is especially interesting when considering the use of a second monitor (second screen) by the user.

A similar feature to measure eye closeness was suggested by Lee, Lee and Park (2010), but it was derived from the eye segmentation in a binary image instead of the palpebral fissure and relied on morphological operations. While monitoring the iris or the pupil could be interesting for detecting blinks, this is less practical for incomplete blinks. EAR has been explored in the context of blink and eye closure detection, as it is robust provided that the landmarks are detected. Occlusion of part of the faces is one of the remaining challenges in the field of computer vision Alashbi (2021) and may be distracting for a landmark detector, as pointed out by Soukupová and Cech (2016a). Part of face can be occluded by masks, as it was common during COVID-19 pandemic, garments, clothes, and accessories (hat, veil, hijab, niqab, . . .) and hands.

One difficulty with EAR is that it varies from person to person and even between the eyes of the same person, so a threshold to determine a complete blink for one subject may misclassify blinks for another. Eye completeness can also differ for both eyes. The same can be said of incomplete blinks.

For the palpebral fissure aspect ratio, this would not be the case for complete blinks, as its value is 0 for completely closed eyes. Detecting incomplete blinks still has the same problem of intra- and inter-person variability, as the palpebral fissure is often asymmetric (DOGANAY et al., 2017) and different from person to person.

As for EAR blink detection, there are some workarounds to this problem. For blink detection, instead of a simple constant threshold, Soukupová and Cech (2016a) have used a support vector machine and a time-window. Changes to the EAR metric were also proposed to incorporated temporal information and per user calibration, as discussed. Except for Fogelton and Benesova (2018), Yin et al. (2022) and Zheng et al. (2022a) methods for incomplete blinks analysis require some sort of calibration per user. Inspired in Maior et al. (2020), the first frames of a video could be used to adjust a threshold based on the palpebral fissure aspect ratio to detect incomplete blinks.

It is valid to note that EAR is an artifact develop with the goal of measuring blinks, based on facial landmarks detectors Soukupová and Cech (2016a). The palpebral fissure dimensions, by the other hand, are a biometric measures already studied in other contexts (medical, orbital surgery, facial plastic surgery, congenital or post-traumatic facial disfigurements). Their analysis by photograph is also not uncommon (VASANTHAKUMAR; KUMAR; RAO, 2013), (BRUMMEN et al., 2021). Analyzing the palpebral fissure height and width directly seems to be, therefore, closer to the medical field.

Providing more than just the eyes or the periorbital region to the model is interesting because there are situations that may be particularly hard even for humans to define the state of the eye, or if the frame is part of a blink or not, without context. If the whole face is available, as in the proposed approach, the facial expression (contracted forehead, mouth opened in scream or smile, periorbital lines in relaxed or contracted state) that is given to the model may help. Using temporal information about the opening and closing time for a blink event, may also fulfill this role, and possibly only the periorbital region could be used. For improved blink analysis performance, though, a combination of methods is likely to perform better. This may be specially interesting when analyzing long videos for research or diagnosis purposes, where real-time is not mandatory.

An eye blink analysis system based in the palpebral fissure aspect ratio can be used together with other methods to ensure an ergonomic use of the computer. Toda, Nakai and Liu (2015) developed a software that, using only the webcam of the user's computer, estimates the distance of the user to the screen through the face area (number of pixels with skin color). The aim is to prevent incorrect posturing of the user's neck, not the precise measurement of the target distance, that would require two or more images (cameras). A pop-up alert message is displayed when the distance is below a set threshold. The tests indicated that the face distance was estimated within 9.42% error in the distance from 25 to 55 cm. Alternatively, the distance between the eyes (for example the distance of the centroids of the palpebral fissures) can be used to roughly estimate this distance.

4 METHODS AND METHODOLOGY

In this section, the generation of a deep learning based segmentation model for the palpebral fissures is described in detail, as well as the steps that both precede and follow its use in a video analysis context to detect complete and incomplete blinks. The Figure 4.1 shows a schematic depicting the use of a palpebral fissure segmentation model to determine eye state.

Figure 4.1: Palpebral fissure segmentation model for eye state determination diagram

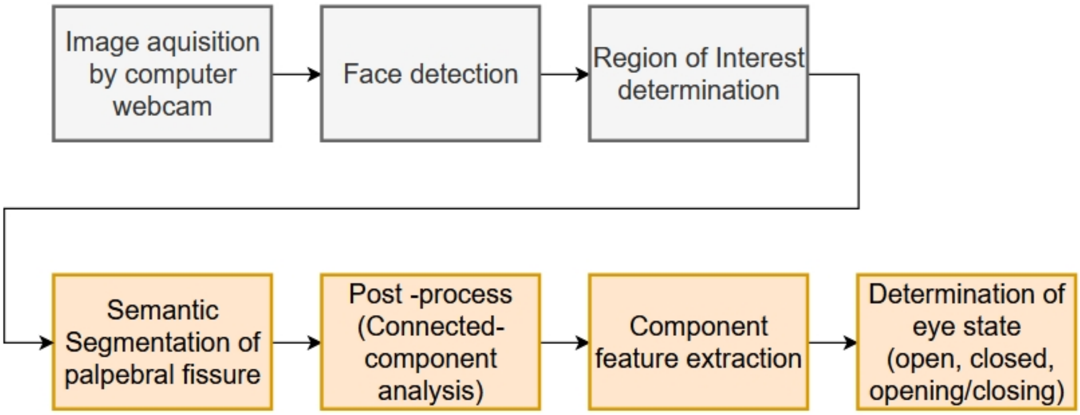
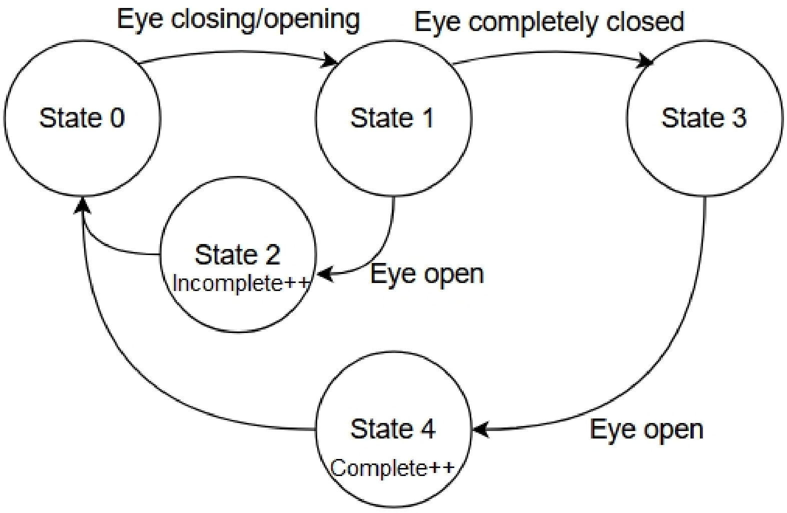


Image aquisition by computer webcam
Face detection
Region of Interest determination
Semantic Segmentation of palpebral fissure
Post -process (Connected-component analysis)
Component feature extraction
Determination of eye state (open, closed, opening/closing)

Source: Author.

The Figure 4.2 presents a simple high-level state machine overview of a complete and incomplete blink-detection system, based on such palpebral fissure segmentation.

Figure 4.2: High-level state machine of blink completeness analysis algorithm



Source: Author.

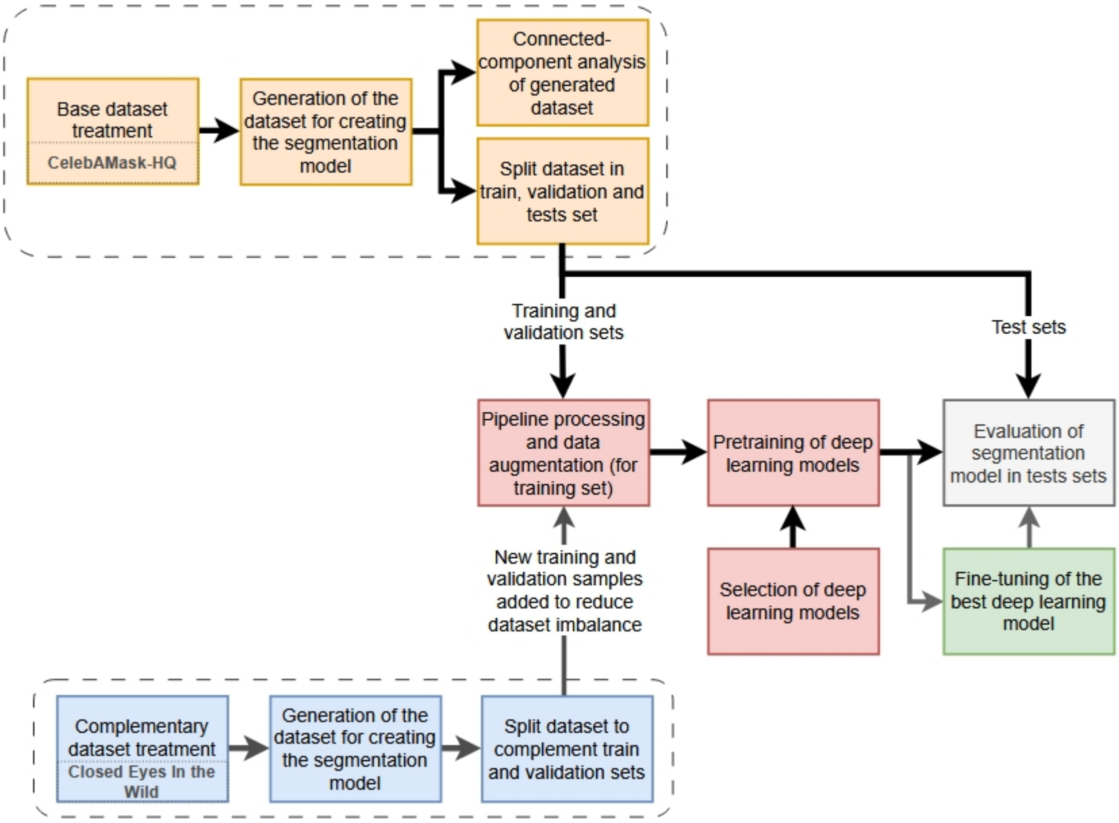
In practice, however, the system could benefit of using the information of previous frames. For example, a sequence composed of a frame for which the prediction is open eyes, closing eyes (or completely closed eyes) and open eyes, respectively, is likely to be caused by a misclassification, because a blink takes more than one frame with an acquisition of 30 frames per second.

Alternatively, the output of the last block in the Figure 4.1 could be the features extracted from the palpebral fissure, such as the ratio of height to width. Similarly to Zheng et al. (2022a), a blink profile, here based on this ratio, could be generated with the proposed segmentation model. The height to width ratio of the segmented palpebral fissure could be used as a feature without the morphological operations of erosion and dilatation used for the eyes in Lee, Lee and Park (2010), only performing the connected-component analysis post-processing step followed by the authors, with up to the two largest areas (excluding the background) been considered as palpebral fissures.

4.1 Overview

To develop the segmentation model, the steps shown in Figure 4.3 are followed. The first steps consist in the analysis and treatment of the base dataset selected for training the segmentation model, the CelebAMask-HQ dataset (LEE et al., 2020), as described in section 4.3 Dataset generation. The dataset is then split in sets for training, validation, and evaluation. These subsets are then processed by a pipeline that applies data augmentation on-the-fly on the training set, as in section 4.4 Pipeline with data augmentation for processing images and mask, preparing the images and segmentations mask to be used as input for training the selected deep learning models, that are discussed in section 4.5 Deep Learning Segmentation Models. Pretraining of the models and hyperparameters tuning is covered in section 4.6 Deep Learning models pretraining phase, while evaluation is covered in section 4.7 Deep Learning models evaluation on images of the generated dataset. Section 4.8 Reducing generated dataset imbalance address the attempt of using images from the Closed Eyes in the Wild dataset to improve the models. Finally, 4.9 Evaluation of the best model after training describes the fine-tuning of the best model.

Figure 4.3: Diagram of the development stages of a palpebral fissure segmentation model for eye state determination


Base dataset treatment
CelebAMask-HQ
Generation of the dataset for creating the segmentation model
Connected-component analysis of generated dataset
Split dataset in train, validation and tests set
Training and validation sets
Test sets
Pipeline processing and data augmentation (for training set)
Pretraining of deep learning models
Evaluation of segmentation model in tests sets
New training and validation samples added to reduce dataset imbalance
Selection of deep learning models
Fine-tuning of the best deep learning model
Complementary dataset treatment
Closed Eyes In the Wild
Generation of the dataset for creating the segmentation model
Split dataset to complement train and validation sets

Source: Author.

4.2 Packages and libraries

The code for training the machine learning and deep learning models was executed in Google Colab, which runs in a cloud Virtual Machine. The Google Colab notebook is running on top of Ubuntu operating system. The packages and libraries used with python 3.10.12 programming language can be seen in Table 4.1:

Table 4.1: Packages and Libraries used

Packages and Libraries	Version	Packages and Libraries	Version
Keras	2.12.0	OpenCV	4.7.0
KerasTuner	1.3.5	Pandas	1.5.3
Keras-applications	1.0.8	Segmentation Models	1.0.1
Image-classifiers	1.0.0	Sklearn	1.2.2
Matplotlib	3.7.1	TensorFlow	2.12.0
Numpy	1.22.4		

Source: Author.

To ensure a reproducible hash, "PYTHONHASHSEED" environment variable was set to 0 in the begging of the executions of code. The random generators of Python, Numpy and TensorFlow also received a "seed" value, to make most of the program fully deterministic, like the shuffling of the dataset and its batches. The training of the models, which are run in GPU, involves certain non-deterministic operations that create sources of randomness.

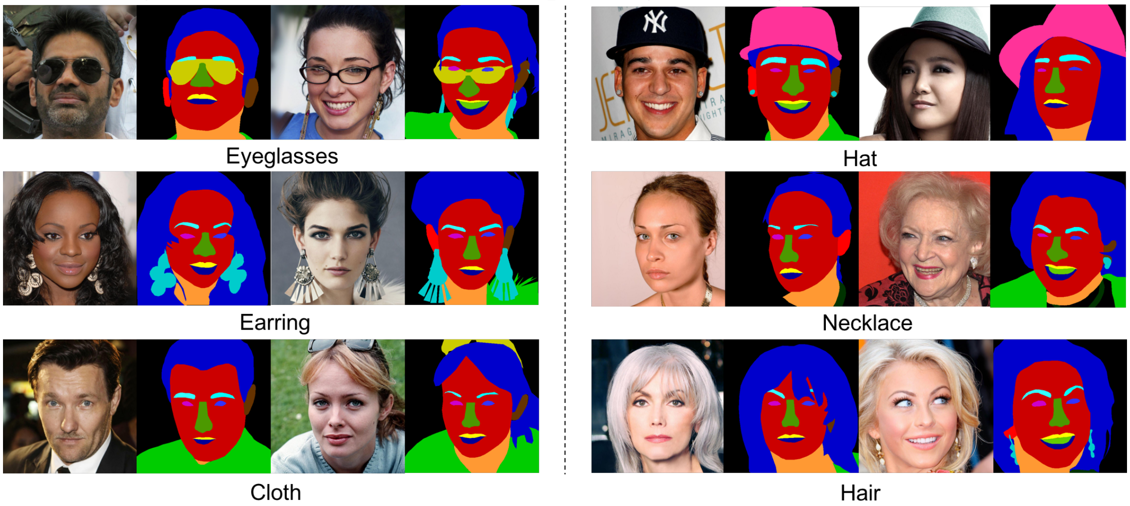
4.3 Dataset generation

4.3.1 Base dataset: CelebAMask-HQ

CelebAMask-HQ dataset (LEE et al., 2020) contains 30000 high-resolution face images (512×512 resolution) and the correspondent segmentation mask of 19 facial attributes and accessories. This is a large scale dataset when compared, for example, to Helen Facial Feature Dataset (LE et al., 2012), which contains 2300 high resolution images also with 11 primary facial components according to Lee et al. (2020).

CelebAMask-HQ dataset images were selected from the CelebFaces Attributes Dataset (CelebA dataset, Liu et al. (2015)) by following CelebA-HQ (KARRAS et al., 2018). Figure 4.4 shows some sample images and segmentations masks of the dataset. Note that each segmentation mask is actually a binary image (black-and-white) file in Portable Network Graphics (PNG) format.

Figure 4.4: CelebAMask-HQ sample images and segmentations masks



Multiple segmentations are displayed together for each image.

Source: Lee et al. (2020).

These masks are identified with a 5-digit numeric prefix (e.g. 00001) followed by the segmentation type, which have the following form:

- *_l_eye.png*, for the segmentation image of the left eye;
- *_r_eye.png*, for the segmentation image of the right eye;
- *_eye_g.png*, for the glasses' segmentation image;

A python script (using python 3.10.11, Numpy 1.22.4, Pandas 1.5.3) was developed to, for each image, indicate in a CVS file these parameters:

- "*number*", the 5-digit numeric prefix for the mask and identifier of the image;
- "*Folder*", one of 14 folders where the segmentation masks are organized;
- "*Left_eye*", 1, if there is a file with the *number* prefix followed by *_l_eye.png* in its name, 0 otherwise;
- "*Right_eye*", 1, if there is a file with the *number* prefix followed by *_r_eye.png* in its name, 0 otherwise;
- "*Eyeglasses*", 1, if there is a file with the *number* prefix followed by *_eye_g.png* in its name, 0 otherwise;
- "*No_eye_segmented*", 1, if there are no files with the *number* prefix followed by with *_l_eye.png* or *_r_eye.png* in their name.

The Table 4.2 shows number of segmentation masks by annotation type.

Table 4.2: Number of segmentation masks by annotation type

Type of annotation	**N of segmentation masks annotated**
left eye	29258
right eye	29260
both eyes annotated	29132
left eye only annotated	126
right eye only noted	128
no eyes annotated	614
glasses noted	1549
no eyes annotated with glasses annotated	406
no glasses annotated nor eyes noted	208

Source: Author.

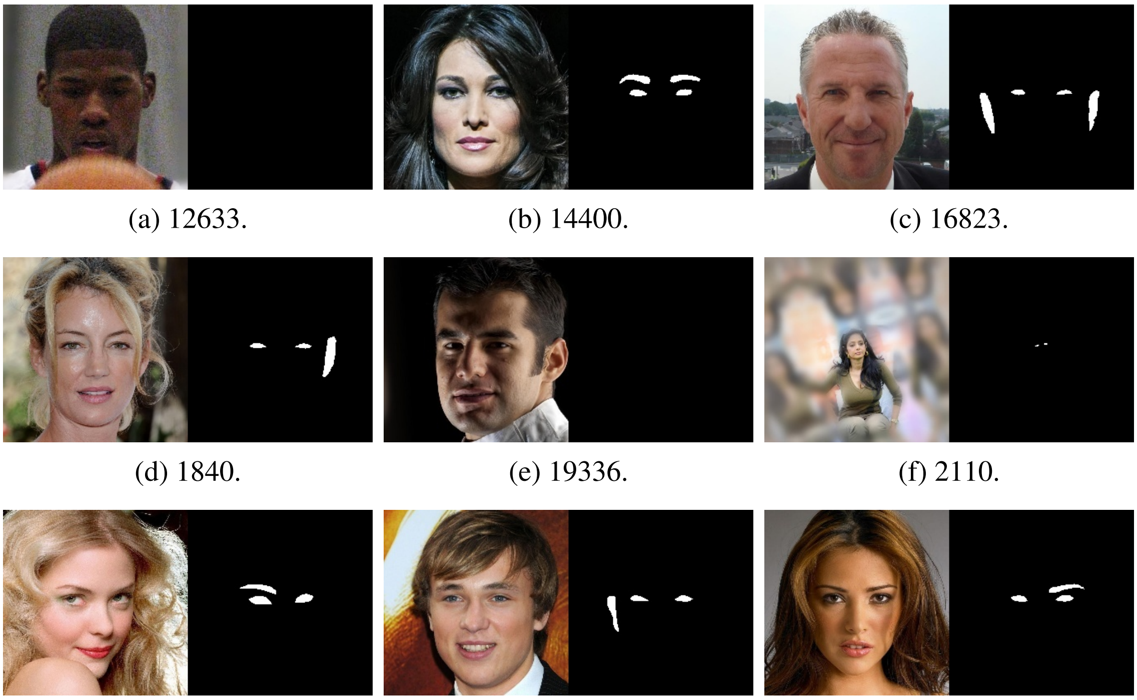
4.3.2 Dataset treatment

Based on the dataset CelebAMask-HQ, a new dataset was generated considering the segmentations masks of the left and right eyes available in the dataset CelebAMask-HQ. If the image does not have these two segmentations masks, the eyes have not been annotated for this image. This can be the case if the eyes are not completely visible (by wearing sunglasses, for example) or are closed.

The right and left eye segmentation masks were combined, generating a single image, using the OpenCV function *add*. Completely black images were generated for the images with no segmentation of either eye.

In addition, a visual inspection was performed on the generated masks. 62 images for which the segmentation presented some kind of problem (e.g., one ear or eyebrow were also annotated) were discarded. The Figure 4.5 shows some discarded samples. The list of discarded images with the identification of the problem that caused the exclusion can be found in the appendices.

Figure 4.5: Discarded samples of CelebAMask-HQ



(a) 12633.

(b) 14400.

(c) 16823.

(d) 1840.

(e) 19336.

(f) 2110.

(g) 2807.

(h) 6136.

(i) 8181.

(a) visible eyes (looking down); (b) eyebrows also annotated; (c) ears annotated; (d) left ear annotated; (e) visible eyes; (f) outlier: small eyes compared to the rest of the dataset; (g) right eyebrow annotated; (h) right ear annotated; (i) left eyebrow annotated.

Source: Author, based on CelebAMask-HQ (LEE et al., 2020).

The Table 4.3 shows the number of segmentation masks by annotation type after the discard of the 62 pairs of images and masks.

Table 4.3: Number of segmentation masks by annotation type after visual inspection

Type of annotation	**N of segmentation masks annotated**
left eye	29205
right eye	29209
both eyes annotated	29086
left eye only annotated	119
right eye only annotated	123
no eyes annotated	610
glasses noted	1545
no eyes annotated with glasses annotated	406
no glasses annotated nor eyes noted	204

Source: Author.

The images where only the left eye or the right eye were annotated ($119 + 123 = 242$ pairs of images and masks) were discarded, as well as the images for which no eyes were annotated and the person was using eyeglasses ("*Left_eye*" and "*Right_eye*" equals to 0 while "*Eyeglasses*" equals to 1; 406 pairs of images and masks). For the later case, in most cases, this indicates that the person was using sunglasses, which is not expected for this indoor application. For the first case, there were images where the person was in profile or one eye was not visible because of the angle between the face and the camera. These images and masks were also discarded because they do not represent the typical situation where a computer user is facing the screen and the eyes are expected to blink together in general.

It is observed that the base dataset has some small inconsistencies in annotation, a natural aspect of annotations performed by multiple individuals, as mentioned by (LEE et al., 2020). Although quality control is mentioned by the authors, some minor inconsistencies occur in the dataset due to subjective interpretations by each annotator, such as an image of an eye containing only pixels of the palpebral fissure or some of its surroundings, such as part of the eyelid. Given the size of the dataset, it is expected that an intelligent model can learn the essence of the data and that these annotations containing a bit more (or less) of the palpebral fissure will not be represented by the model.

The segmentation masks generated and the original images were then resized to 224 x 224 using *resize* of OpenCV with bilinear interpolation (INTER_LINEAR). This reduction in the input image resolution allows a memory usage reduction. 224 is the standard input shape of some deep learning models pretrained in *ImageNet*, like the Keras implementations of VGG (SIMONYAN; ZISSERMAN, 2014) and MobileNetV2 (SANDLER et al., 2018). For the latter architecture, when the input shape is undefined or non-square, or square but not in [96, 128, 160, 192, 224], weights for input shape (224, 224) are loaded as the default. The images and segmentation masks were then compressed to be used in Google Colab platform.

4.3.3 Connected-component analysis

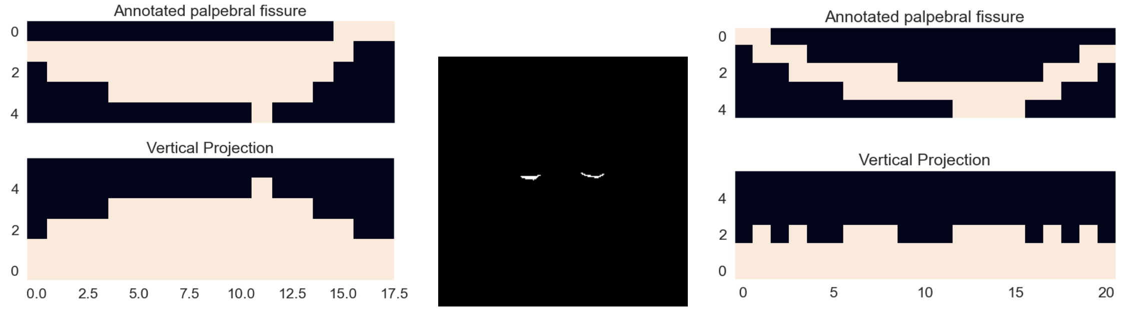
A connected-component analysis (CCA) is performed in the generated masks where both eyes are open. The goal is to analyze the dimensions and coordinates of the palpebral fissures. The dimensions are useful to get insights on values that can be used as a threshold when both eyes are completed close, but the mask has some pixels set (instead of an empty mask). The extreme coordinates are useful to establish reasonable translation values to the data augmentations performed in the images, to avoid cropping the eyes. For the topmost and bottom most points, the y coordinate is relevant for determining the vertical translation. For the horizontal translation, the x coordinate of the leftmost point of the right palpebral fissure and the rightmost point of the left palpebral fissure are of interest.

CCA is the first step in this process. OpenCV 4.7.0 implementation of Spaghetti algorithm (BOLELLI et al., 2020) for 8-way connectivity, *connectedComponentsWithStats*, is used to compute the connected components labeled in the segmentation masks. The components are then filtered with only the two largest areas detected being considered. A minimal size area equals 0 is used for filtering when analyzing the dataset.

The next step is to find the contours of the two largest palpebral fissures. A topological structural analysis algorithm for binary images (OpenCV 4.7.0 implementation of Suzuki and Abe (1985), *findContours*) is used to perform this.

With the contours of both palpebral fissures, the bounding box containing each palpebral fissure can be computed. This gives the width of the palpebral fissure. The height is computed using the maximal value of the vertical projection profile along the y axis of the palpebral fissure (i.e, for every mask column, the sum of all column pixel values is computed; the maximal value is the height). Using the height of the bounding box directly does not fully represent what is done in the EAR feature: a closing eye in a waning/waxing crescent moon shape would have a bigger height, as exemplified in Figure 4.6.

Figure 4.6: Segmentation mask with one eye in a waning/waxing crescent moon shape



(a) Annotated right palpebral fissure and vertical projection profile, with maximal value of 4 pixels.

(b) Generated image segmentation mask.

(c) Annotated left palpebral fissure mask and vertical projection profile, with maximal value of 2 pixels.

Source: Author, based on CelebAMask-HQ.

The extreme points are computed individually from the dimensions. Only the coordinates relevant for the axis in question should be considered, because when multiple pixels have the same coordinate, the first occurrence found is chosen. For the leftmost point, if several pixels have the same x coordinate, the first one in the contour list is chosen, so the y coordinate may not be the most representative for the eye corner. The same goes for the bottom most and the top most point: only the y coordinate is representative: the x coordinate may not be centralized in the palpebral fissure, as in Figure 4.6c.

For the extreme coordinates of the palpebral fissure, what is significant is:

- the x value of the left most and right most pixel;
- the y value of the top most and bottom most pixel.

4.3.4 Splitting the dataset

The sample indexes (images and corresponding masks) were randomized and divided into training, validation, and test sets in the ratio 6:2:2. This was done using stratified sampling, to ensure that the relative frequency between samples annotated as having both eyes open and both eyes completely closed in the dataset were approximately preserved in each set. The original dataset is highly imbalanced in this aspect.

Two subsets with of the test set were also defined. The "closed eyes test set" contains the closed eyes samples, while the "opened eyes test set" contains the rest of the samples of the test set (both eyes opened). They allow a visual verification of the predictions and that the imbalance of the class in terms of images with both eyes closed is not misleading the models in the segmentation task.

4.4 Pipeline with data augmentation for processing images and mask

A data pipeline with data augmentation "on-the-fly" was written based on the *Dataset* class of the *TensorFlow's data* module to feed the samples to train the models. The images (JPEG format) and masks (PNG format) are supplied in batches to the model in order to reduce the use of RAM. Also, while the model uses some samples for training, through *prefetching* another batch is prepared at the same time.

The samples are read from the disk and decoded. The masks are scaled by a factor of $\frac{1}{255}$, so the pixel values are 0 to 1 later on. Samples are then cached in memory the first time the dataset is iterated over. This save some operations (like file opening and data reading) from being executed during each epoch. Training samples are also pseudorandomly reshuffled for each epoch (after retrieving data from cache) and the batches are constituted after shuffling to get unique batches for each epoch. For the training set, data augmentation is applied on a batch of items at once, thus vectorizing the mapping, to reduce the overhead related to scheduling and executing augmentations.

4.4.1 Reading and resizing JPEG images in different frameworks

As discussed in Sinha (2020), there are differences in a JPEG image read and resized between frameworks, for example, OpenCV, TensorFlow, and Pillow. These steps may impact the pre-processing of the images. As OpenCV is focused on real-time applications and has modules to interface with the camera of a computer, a verification of the type of JPEG decompression by TensorFlow, which is used to develop the deep learning modules, is performed in Google Colab to ensure compatibility.

The images read are decoded using the function *tensorflow.io.decode_jpg*; the argument *dct_method* equals to *INTEGER_ACCURATE* is used to specify a hint about the algorithm used for JPEG decompression. The type hint was respected for the experiments described here, and a model trained with this pipeline should not be affected when changing the reading function of JPEG images from TensorFlow to OpenCV.

4.4.2 Data augmentation on-the-fly for training set

As the images in the CelebAMask-HQ dataset are most of the time bright, an adjustment is randomly applied to reduce their brightness by a factor. This happens if a random sorted value in the half-open interval $[0.0, 1.0)$ is less than 0.5, and the factor is 0.4 times this random value, so the brightness reduction is of 20% at most.

The images and mask pairs are randomly flipped horizontally (left to right), and translation from -20% to 20% in the horizontal axis and from -25% to 25% in the vertical axis are also applied. The same data augmentations are applied to all images and masks (for translation and flipping along vertical axes) in an epoch, which will be processed asynchronously on the CPU. The data therefore given to the model is likely to be different each epoch.

Empty space due to the translations is filled with zeros instead of other types of filling. This is based in the work of Hashemi (2019), that proposed zero-padding around smaller images, as opposed to interpolation, to resize images to a fixed size before passing then to a CNN. The author has shown that zero-padding has no effect on the classification accuracy, but reduced the training time.

The vertical translations are important because a face detection algorithm, like OpenCV's implementation of Viola-Jones algorithm (VIOLA; JONES, 2004), may determine a bounding box containing only part of the face. Eyeblink8 annotations, for example, don't consider the whole face, as many times forehead and hair is missing; CelebAMask-HQ images, on the other hand, contain many "complete" faces, with even some surrounding features.

4.5 Deep Learning Segmentation Models

Two models architectures are tested for the binary image segmentation: Unet and Linknet. The backbones used for the encoder path are 'resnet18' and 'mobilenetv2' for each architecture. All backbones have pre-trained weights on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset for faster and better convergence. Initializing the models with the pre-trained weights on ILSVRC 2012 helps the model to already recognize not only curved lines and shapes like circles and ellipsis, but also human palpebral fissures.

The python library *Segmentation Models* (IAKUBOVSKII, 2019) with Neural

Networks for Image Segmentation based on Keras and TensorFlow is used to define the deep learning modules and to benefit of transfer learning.

The backbones of *Segmentation Models*, as each Keras Application models, expect a specific kind of input preprocessing. This is done using *Segmentation Models*'s *get_preprocessing* method with the name of the backbone as argument. For MobileNetV2 model, this corresponds to scale input pixels between -1 and 1. For ResNet, input images are converted from RGB color space to BGR, then each color channel is zero-centered with respect to ImageNet dataset, without scaling.

The preprocessing layers are part of the model that is going to be trained. According to the TensorFlow documentation (CHOLLET; OMERNICK, 2021), doing preprocessing inside the model has some benefits: it benefits from GPU acceleration; and, at inference time, this option makes the model portable and it helps reduce the training/serving skew (difference between performance during training and performance during serving). It also avoids reimplementing the pipeline when exporting the model to another runtime, such as TensorFlow.js.

Batch normalization is not a part of the original Unet design, as it was introduced latter. Here, batch normalization is used because of its ability to improve convergence and reduce training time.

4.6 Deep Learning models pretraining phase

While the encoders have pre-trained weights, the decoders are randomly initialized. The pretraining phase corresponds to adjust the decoders weights values. As stated in Iakubovskii (2018), "sometimes, it is useful to train only randomly initialized decoder in order not to damage weights of properly trained encoder with huge gradients during first steps of training". This may lead to a performance that may be enough for the application, without a need for fine-tuning (unfreezing all or some layers of the encoder and training).

As a first approach for the pretraining, the hyperparameters are set manually, as explained in subsection 4.6.1 Manual setting of hyperparameters. As mentioned previously, searching the hyperparameter space is relevant to obtaining the best score, or at least verifying that the previous manual choice has led to a good performance in the sample space. This is described in subsection 4.6.2 Searching the hyperparameter space.

4.6.1 Manual setting of hyperparameters

A combined loss is used as a cost function, as done, for example, in Kolarik, Burget and Riha (2020) and in a well ranked solution to Carvana Image Masking Challenge in Kaggle by Yokoo (2017). The loss is 1 minus the value of the Dice coefficient calculated per image (loss is calculated for each image in batch and then averaged) increased by the binary cross-entropy, as expressed in Equation 4.1.

$$bce\_dice\_loss_{\text{per batch}} = \left(1 - \frac{1}{B} \sum_{i=1}^{B} \underbrace{\frac{(1+\beta^2) \cdot TP_i + \varepsilon}{(1+\beta^2) \cdot TP_i + \beta^2 FN_i + FP_i + \varepsilon}}_{\text{Dice per image}} \right) + BCE \quad (4.1)$$

where

- B - number of samples in batch;
- β - f-score coefficient; equals to 1 for the Dice coefficient;
- TP_i - true positive pixels for mask i;
- FP_i - false positive pixels for mask i;
- FN_i - false negative pixels for mask i;
- ε - smooth coefficient to avoid division by zero, equals to $1e^{-05}$;
- BCE is defined in Equation 2.4.

Cross-entropy prioritizes the overall pixel-wise accuracy, treating each pixel as an independent prediction, while Dice loss acts in the level of the resulting mask and is less sensitive to class imbalance (MU; SUN; HE, 2022).

The loss, as well as the Dice similarity coefficient and Jaccard's index (Intersection over Union) are monitored during model training. The loss functions and performance metrics use the implementation available in the python module *Segmentation Models*, adding a smooth coefficient to the numerator and to the denominator to avoid division by zero. The default value for both Dice coefficient and Jaccard index is $1e^{-05}$.

The Adam optimization algorithm (KINGMA; BA, 2017) is used as an optimizer of the loss function, which is an extension to stochastic gradient descent method and is based on adaptive estimation of first-order and second-order moments. The Keras implementation is used with default parameters (β_1, the exponential decay rate for the first moment estimates equals to 0.9; β_2, the exponential decay rate for the second moment estimates, equals to 0.999; and ε, a constant for numerical stability, equals to $1e-07$).

For this experiment, the learning rate is equal to 0.0015 (divided by 5 after the epochs 10 and 15) and the regularization factor L2 is equals 0.005 for each model. Pretraining is performed over 20 epochs, with the best model obtained so far based on the maximal Dice coefficient of the validation set being saved. This type of early stopping technique adds a regularization effect to the pretraining, reducing the magnitude of overfitting. The batches contain 32 samples, to reduce large variances in training batch normalization layers caused by smaller mini-batch sizes. Three experiments for each combination of architecture and backbone are realized to reduce the variance of the results.

4.6.2 Searching the hyperparameter space

The Keras Tuner library O'Malley et al. (2019) implementation of Random Search was used for searching the hyperparameter space for the learning rate and L2 regularization factor. Table 4.4 shows the minimal and maximal values used.

Table 4.4: Minimal and maximal values for hyperparameter space

	Minimal value	Maximal value
Learning Rate (LR)	0.0001	0.001
L2 regularization factor (λ)	0.01	0.1

Source: Author.

Log sampling value was used, according to Equation 4.2.

$$\text{hyperparameter value} = \text{minimal value} \cdot \left(\frac{\text{maximal value}}{\text{minimal value}} \right)^{value} \tag{4.2}$$

where value is in the range $[0.0, 1.0)$. The minimal and maximal limits are arbitrated. The maximal value of learning rate in the distribution is smaller than the value used before (0.0015). Also, the maximal regularization in its interval is greater than the value used before (0.005). This is because the networks are likely to present some degree of overfitting, given the complexity of the networks, and both increasing the L2 regularization as decreasing the learning rate may reduce overfitting.

Two random searches are performed for each combination of architecture and backbone. The first one consists of 15 trials, with each trial having different hyperparameter values and being executed once. The idea is to quickly explore the hyperparameter search space. The Dice metric of the validation set is used directly as objective, and the optimizer tries to maximize it. The best combination of hyperparameters is then used to train each model in the same fashion as the first manual test (3 times). The second random search consists of 8 trials with 3 executions per trial, with each execution within the same trial having the same hyperparameter values. The purpose of having multiple executions per trial is to reduce the variance of the results and therefore be able to more accurately assess the performance of a mode. The objective to be minimized is the combined loss function of the validation set. The models obtained in both random searches are then evaluated in the test set.

Searching in a low-dimensional space is often done with grid search, which is less practical in high-dimensional spaces, which is not the case here. Because the importance of each parameter was not known a priori, random search was chosen to explore the hyperparameter space. Furthermore, random sampling helps find good candidates faster.

4.7 Deep Learning models evaluation on images of the generated dataset

The evaluation, prediction, and evaluation of time for prediction are executed in the test set, to provide an unbiased evaluation of model fit and verify if the model generalizes well to new data.

For the evaluation, the Dice coefficient score, IoU score as well as the loss, are computed. Zheng et al. (2022a) considered a Dice similarity coefficient of a model over 90% to be reliable for the segmentation model used to segment the palpebral fissures obtain from the Keratograph 5M. This is adopted as the base criteria.

For visualizing the prediction masks, a figure is generated for each pair of 20 samples of the test set, containing 6 subfigures: the original mask segmentation; an overlay between the original and the generated mask; the generated mask; an overlay between the original mask and the image; the original image; and an overlay between the generated mask and the image. A connected-component analysis followed by the contour detection to determine palpebral dimensions are also performed. The same visualization procedure is also applied to 20 samples of the closed eyes test set.

The time for inference per image in GPU is evaluated in 30 batches in 10 repetitions, each batch with 32 images. Another test is performed with the 960 images not predicted in batches, using *predict* and *__call__* methods of Keras models, again with 10 repetitions. The idea is to simulate directly processing an image retrieved from a video.

Two confusion matrices and a histogram are also used to determine the ability of the models to distinguish between two open and two fully closed eyes in faces. The first confusion matrix assumes that if at least one palpebral fissure is not fully closed (i.e., at least one connected component was found), then the eyes are not fully closed. As in a spontaneous blink, both eyes typically close, a second confusion matrix considers that if at least one palpebral fissure is fully closed, the eyes are closed. In terms of segmentation mask prediction, this means that the mask is not empty but has only one connected component detected. In practice, a good model can present a segmentation mask with only a few incorrect pixels instead of an empty mask when both eyes are closed. As the two biggest connected components are not filtered when computing the confusion matrices, a histogram of the maximal area determined in the CCA is also displayed.

4.8 Reducing generated dataset imbalance

With the goal of investigating the impact of reducing the imbalance of the dataset, face images with closed eyes from the Closed Eyes in the Wild (CEW) were added to the training and validation sets. CEW dataset, presented by Song et al. (2014), contains approximately 4800 images, between left and right eyes, closed and open, taken from the Labeled Face in the Wild database (HUANG et al., 2007). In particular, there are 1192 face images containing closed eyes. Some challenges of this set include amateur photography, occlusions, lighting problems, pose, and motion blur, as noted by Alparslan, Alparslan and Burlick (2020). Figure 4.7 shows some samples.

Figure 4.7: Closed Eyes in The Wild sample images

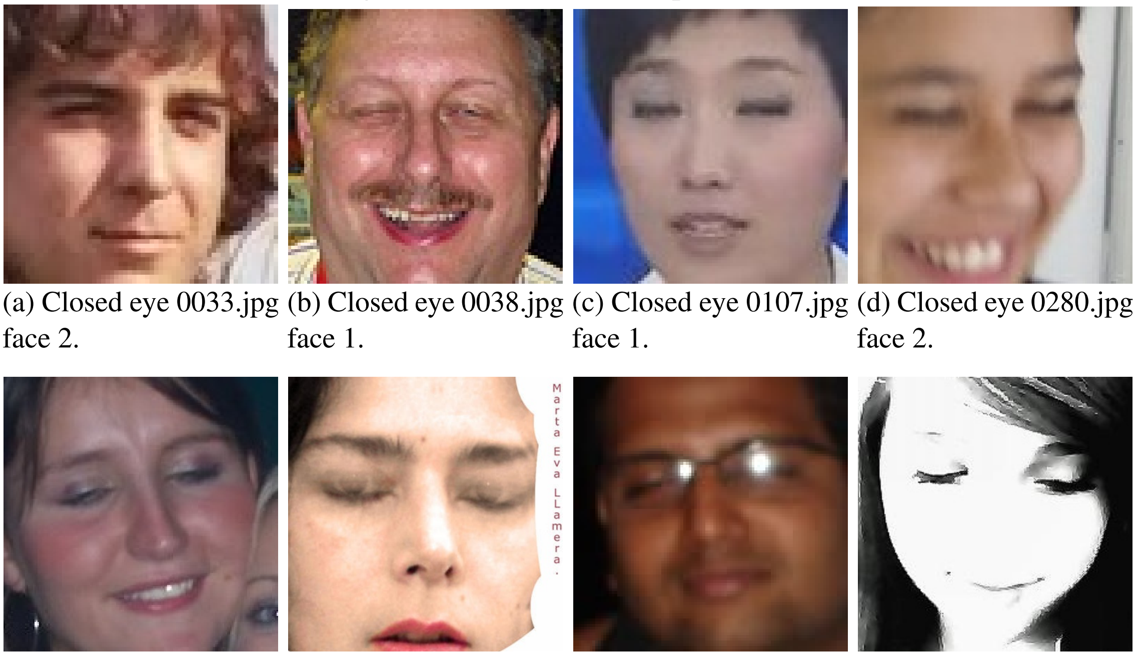
Source: Song et al. (2014).

4.8.1 Dataset treatment

CEW consists of images with different sizes, not always with square shape (by one pixel line, usually). Images were resized to 224 x 224 for compatibility with the training and validation sets. To keep the aspect ratio of images, they were rescaled such that the shorter side was of length 224, and then cropped keeping the top left side unchanged. The generated masks are empty, as only faces with completely closed eyes were selected.

The CEW dataset was filtered, so that only faces with both eyes completely closed and with a reasonable resolution were used. In a first pass through the dataset, images with partially open eyes, eyes from another person or situations where it was difficult to judge the state of the eye due to low resolution or blur were marked. Then, all images with a minimal size lower than 73 were also marked. The marked images were then excluded. The Figure 4.8 shows some examples of the 365 discarded samples.

Figure 4.8: Discarded samples CEW



(a) Closed eye 0033.jpg face 2.
(b) Closed eye 0038.jpg face 1.
(c) Closed eye 0107.jpg face 1.
(d) Closed eye 0280.jpg face 2.
(e) Closed eye 0347.jpg face 2.
(f) Closed eye 0493.jpg face 1.
(g) Closed eye 1263.jpg face 1.
(h) Closed eye 2205.jpg face 2.

(a) partial visible eyes; (b) one partial open eye; (c) low resolution, possible 1 partial open eye; (d) low resolution; (e) one open eye from another person; (f) partial open eyes; (g) reflection; (h) image effects, face features not clear.

Source: Closed Eyes in the Wild (SONG et al., 2014).

The exclusion of low dimension images is relevant, as resizing an image from 56 x 56 to 224 x 224 would make it specially blur, making it too different from the CelebAMask-HQ dataset (LEE et al., 2020) dataset. This would constitute a sort of bias (TORRALBA; EFROS, 2011) that the model could lean on without improving overall metrics in test sets.

The CEW dataset was then split. 620 CEW samples were randomly assigned to increment the training set, and 207 were assigned to increment the validation set (keeping the ratio of 0.75 between the sets).

4.8.2 Testing the best pretrained models

The model considered the best in terms of dice metric were pretrained and tested again (3 times), now with the increased training and validation sets, considering the same procedure described in subsection 4.6.1 Manual setting of hyperparameters.

4.9 Evaluation of the best model after training

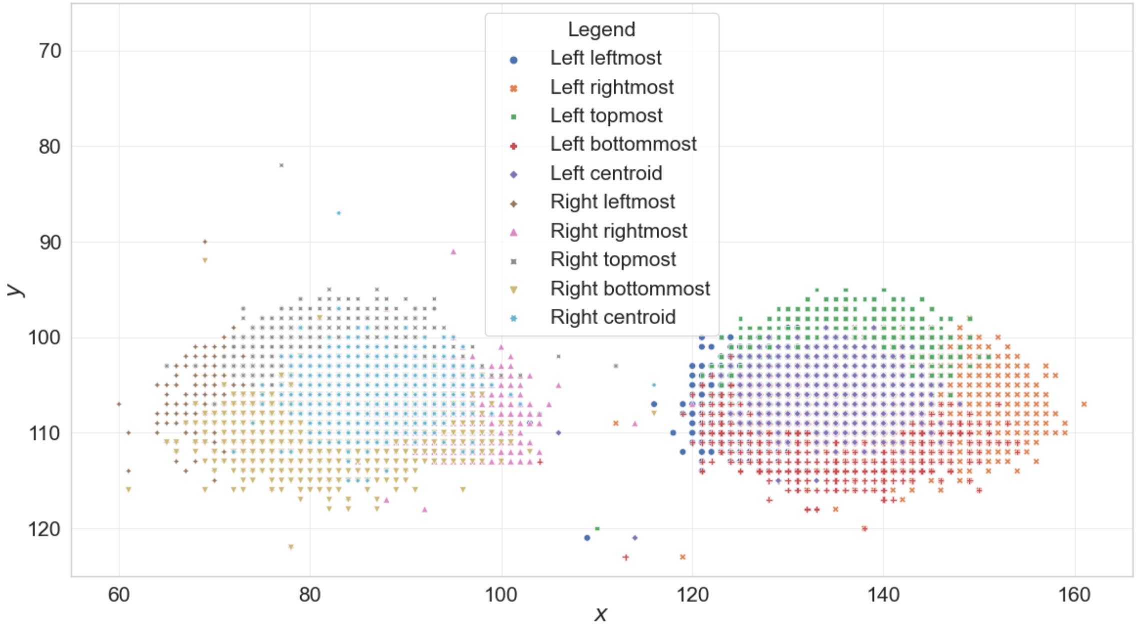
The model considered the best in terms of dice metric is then trained and tested again, with the corresponding training and validation set (3 times). For the training, the last 2 blocks that compose the encoder are unfrozen, and the model is trained for 10 epochs. Then, another 2 blocks that compose the encoder are unfrozen, and the model is trained for another 10 epochs.

5 RESULTS AND DISCUSSION

5.1 Analysis of images with both eyes open for the generated dataset

The scatter plot of palpebral fissure coordinates in Figure 5.1 indicates the position of the coordinates of the palpebral fissure after the contour analysis. As it can be seen, the centroids are usually vertically centralized in images. The leftmost *x* coordinate for the right eye is 60 and the rightmost *x* coordinate for the left eye is 161. As the image size is 224, a 20% translation to left or right (about 45 pixels) in the data augmentation process should not be too aggressive, as it will not position any part of the eye outside the image. Similarly, the topmost *y* coordinate is 82, and the bottom most *y* coordinate is 123, so a vertical translation of 25% (56 pixels) also maintains the eye region in the image.

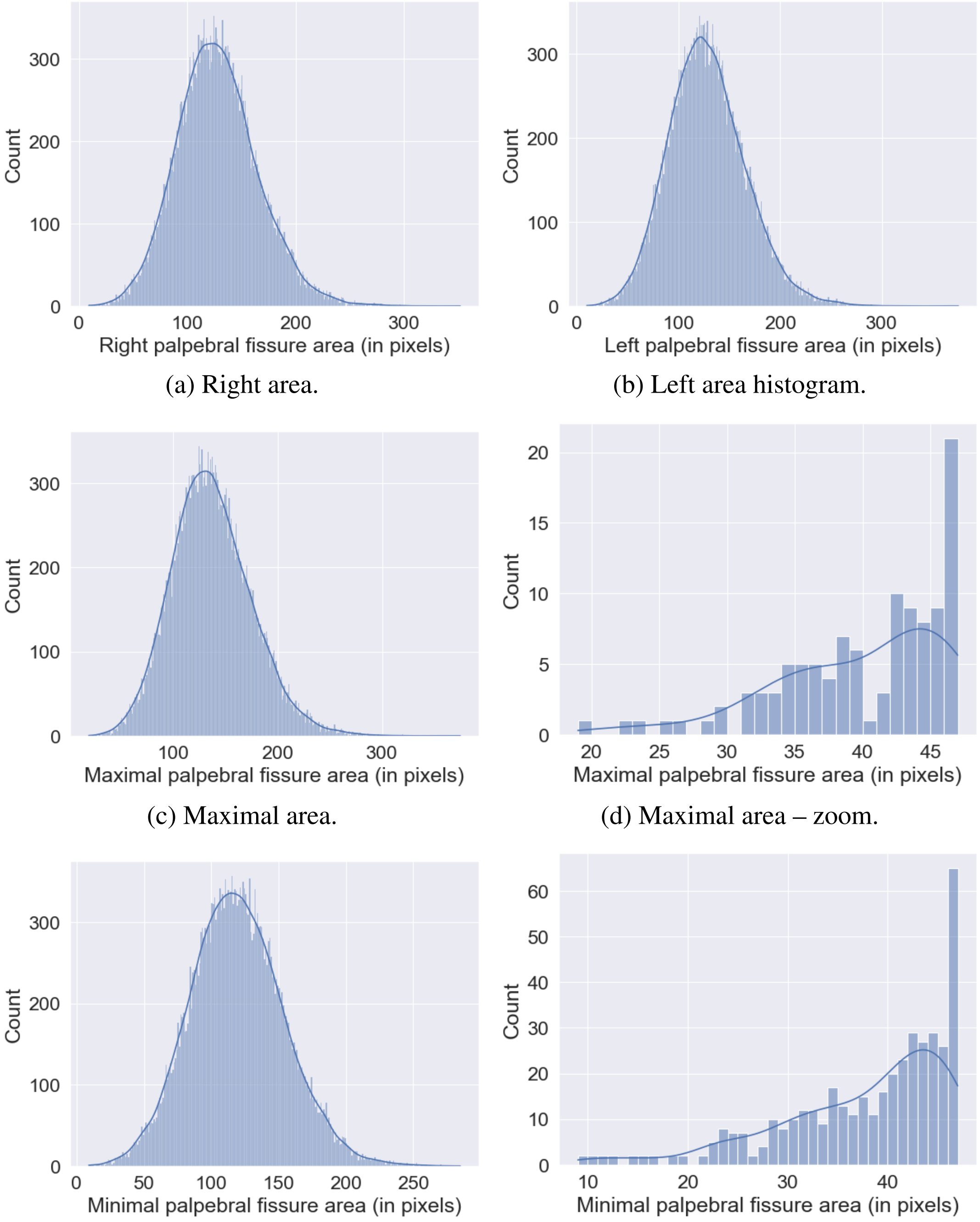
Figure 5.1: Scatter plot of palpebral fissure coordinates



Source: Author.

The Figure 5.2 shows the palpebral fissure area histograms for the 29086 masks with both eyes open. The first aspect is the bell-shaped distribution of the left, right, maximal, and minimal areas. The blue line in the histograms is a kernel density estimate computed to display a smoothed version of the distribution. The mean ± standard deviation for the left and the right palpebral fissures are 128.2 ± 37.8 and 127.9 ± 37.4 pixels, respectively. The mean ± standard deviation for the maximal and minimal palpebral fissures are 136.7 ± 38.3 and 119.4 ± 34.7 pixels, respectively, which illustrates the facts of palpebral fissure asymmetry and that not all images have the person facing the camera.

Figure 5.2: Palpebral fissure area histograms for masks with both eyes open



(a) Right area.

(b) Left area histogram.

(c) Maximal area.

(d) Maximal area – zoom.

(e) Minimal area.

(f) Minimal area – zoom.

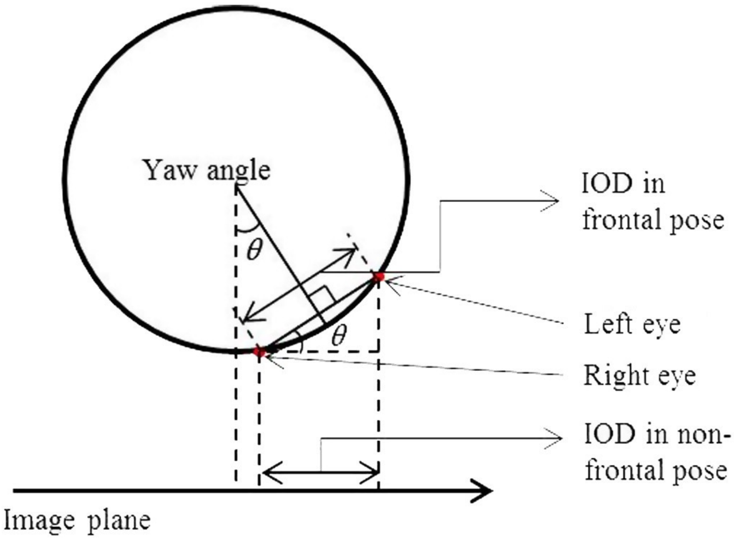
Source: Author.

The Figure 5.2f analysis suggests that ignoring connected-components with small areas (inferior to about 10 pixels of the total, 224 x 224 = 50176) can be used as part of a heuristic for detecting closed eyes even when there is a small error in the segmentation mask prediction. Similarly, as both eyes blink together in a spontaneous blink, Figure 5.2d analysis suggests that ignoring connected-components when the larger one is smaller than a certain small threshold (as about 20 pixels from 50176) can also be considered.

As a reference, Figure 4.5f, an outlier image that was not considered in this analysis because the face covers too little from the image compared to the majority of samples, has two palpebral fissures of areas 8 and 7 pixels. Normally, computer users tend to be too close from the screen. The face detected in the frame is not likely to be as small as the image in Figure 4.5f. This is also likely the case for a frame capture by the front camera of a mobile device.

The eye region area in an image depends on the camera resolution and the distance between the person and the camera, with the size of the eye having a correlation with the intraocular distance. This is, however, affected by yaw angle variations (horizontal non-frontal head rotations), as shown in the Figure 5.3. The same is valid for pitch rotation (vertical non-frontal head rotations, when the person is looking up or down).

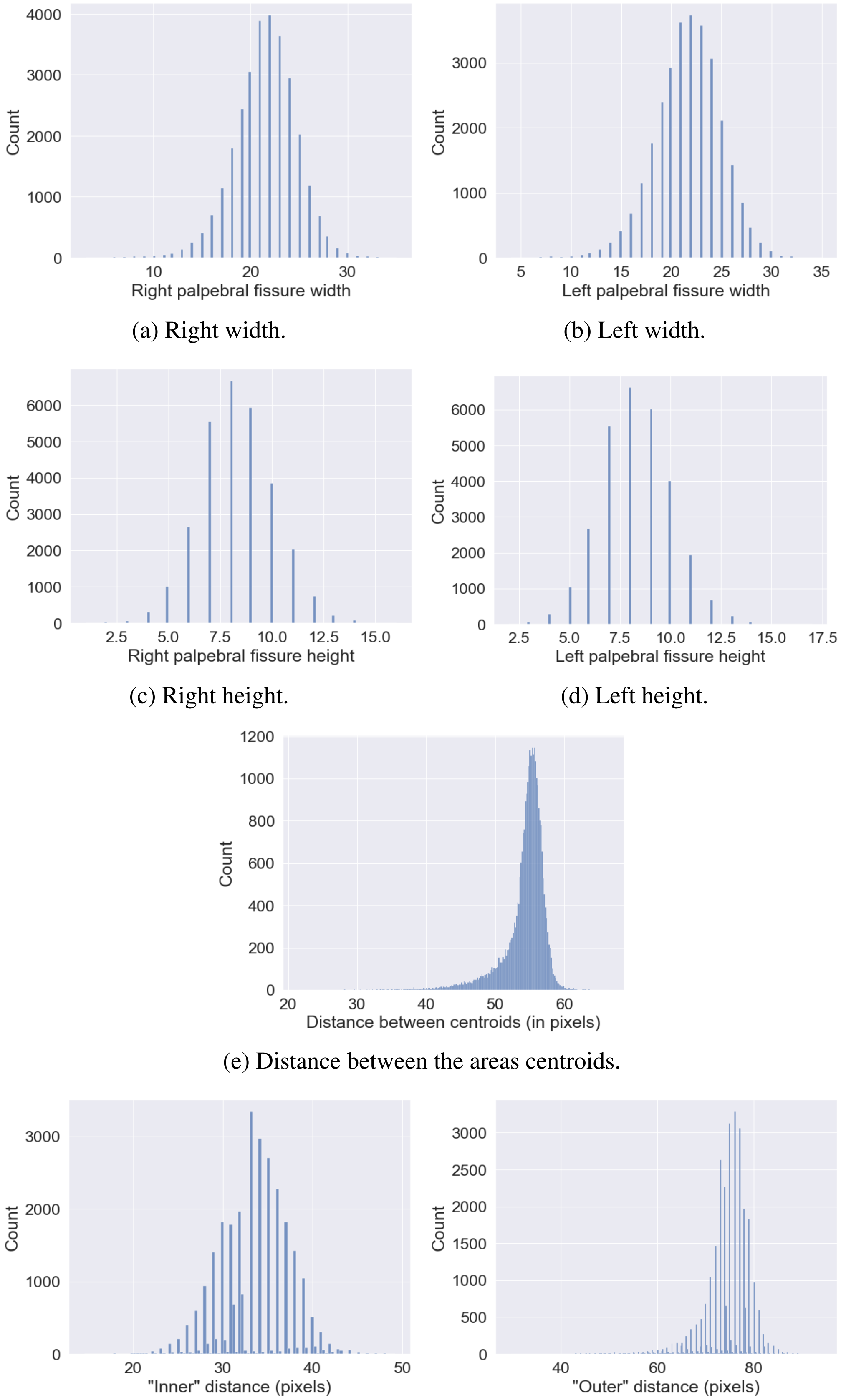
Figure 5.3: Interocular distance (IOD) for pose variation



Source: Adapted from Kim et al. (2017)

To illustrate some samples of CCA and contour detection with usual and extreme values in the dataset, the reader is referred to Appendix B Examples of connected-component analysis of images with both eyes open for the generated dataset. The Figure 5.4 shows the palpebral fissure width and height, as well as distance between palpebral fissures for masks with both eyes open. Note that the inner distance roughly corresponds to the intraocular distance, as the extremes of the palpebral fissures are not always well define for flat regions, as discussed early in subsection 4.3.3 Connected-component analysis. The same consideration applies to the outer distance. All distances are subjected to in-plane rotation of the face (not common in CelebAMask-HQ dataset) and non-frontal head rotations (common), making them of limited use.

Figure 5.4: Palpebral fissure measures histograms for masks with both eyes open



(a) Right width.

(b) Left width.

(c) Right height.

(d) Left height.

(e) Distance between the areas centroids.

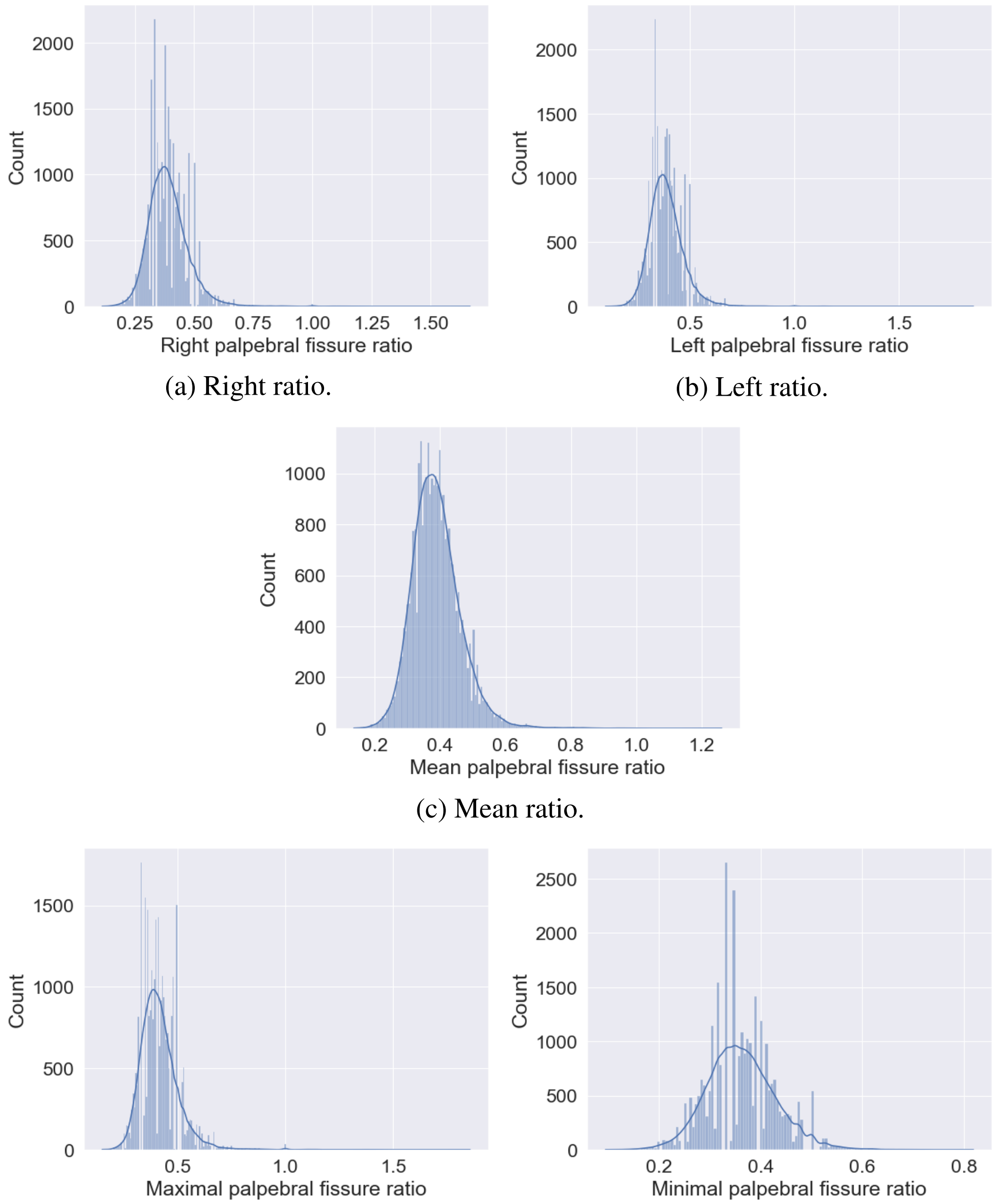
(f) Approximation of the inner distance between extreme points (close to the nose).

(g) Approximation of the outer distance between lateral extreme points.

Source: Author.

The palpebral fissures are not always aligned in CelebAMask-HQ Dataset, which implies that the ratios ("height" by width) computed may not always be meaningful regarding eye closure by themself. The Figure 5.5 shows the palpebral fissure ratio histograms for the masks with both eyes open and makes clear this fact.

Figure 5.5: Palpebral fissure ratio histograms for masks with both eyes open



(a) Right ratio.

(b) Left ratio.

(c) Mean ratio.

(d) Maximal ratio.

(e) Minimal ratio.

Source: Author.

The version used for palpebral fissure aspect ratio is not invariant to in-plane face rotations like the EAR based in landmarks proposed by Soukupová and Cech (2016b). One could find the minimal bounding box containing the eyes to compensate for this angle, but: pure in-plane rotation like in Figure B.7 are rare in the CelebAMask-HQ dataset; and CelebAMask-HQ samples may have strong non-frontal head rotations, with some faces almost in profile (even when images with only both eyes open are considered).

These images with strong face rotations are still relevant to train a segmentation model. For faces where the angle θ in Figure 5.3 is small, and thus cos θ is close to 1, there are no in-plane face rotations, as usually is the case in front of the computer, the version of the palpebral fissure aspect ratio based on the horizontal width of the eyes and the vertical projection is suitable. In-plane face rotations are not usual when using the computer, as a typical user using only one screen usually has the eyes aligned with the horizontal level (i.e., no in-plane rotation), and the head yaw and pitch angles are small (inferior to about 10° for some notebooks). Occupational Safety and Health Administration (n.d.) indicates that monitors should not be farther than 35° degrees to the left or right. When using a bigger screen, or a second screen next to the first one (that is facing the user and where the webcam is positioned), the yaw angle can increase to values greater than 35°. To illustrate the effects of head position in palpebral fissure width and provide a rough idea of head pose angle, the reader is referred to Appendix C Width (in pixels) of the palpebral fissure multiplied by the cosine of the yaw angle.

5.2 Enhancements: palpebral fissure dimensions correction and automatic detection of segmentation errors in static and video images

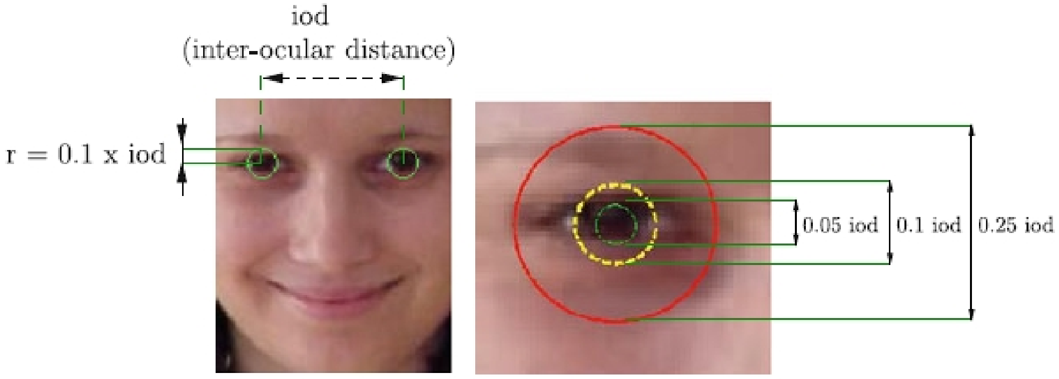
It is possible to estimate the yaw angle θ. Lee, Lee and Park (2010) have used the distance between the eyes and the laterals of the face to empirically estimate the rotation angle of the face. The palpebral fissure dimensions can then be "corrected" for non-frontal faces. Authors have also considered only the largest palpebral fissure available when analyzing blinks in non-frontal faces.

Some characteristics of the palpebral fissure and the face can also be used to automatically detect cases of incorrect segmentation. The face width and height are usually known, as a face detector is used and its output is the input for the model.

Assuming that the user is facing the camera (or the screen and the webcam), if two components have been selected as palpebral fissure candidates, the interocular distance can be verified. An incorrect segmentation is likely to has occurred if the value is "too small": in an average face, the distance between the inner eye corners is roughly equal to the width of one eye (JESORSKY; KIRCHBERG; FRISCHHOLZ, 2001). The use of test-time augmentation or meta-classifiers can help achieve the correct segmentation. The palpebral fissure aspect can also be used to perform this check. Its height is less than its width, so the expected ratio value is smaller than 1. When applying the model to video, the interocular distance can be monitored and compared with previous values, making it easier to detect anomalous segmentations.

While working if eye localization, Ahmad et al. (2022) considered that the distance between the eyes candidates (and here the palpebral fissure candidates) should be within about $1/5$ and $4/5$ of the face size. The Figure 5.6 illustrate the relationship between the area of the eye relative to interocular distance.

Figure 5.6: Eye relative to interocular distance



Source: Monzo et al. (2011).

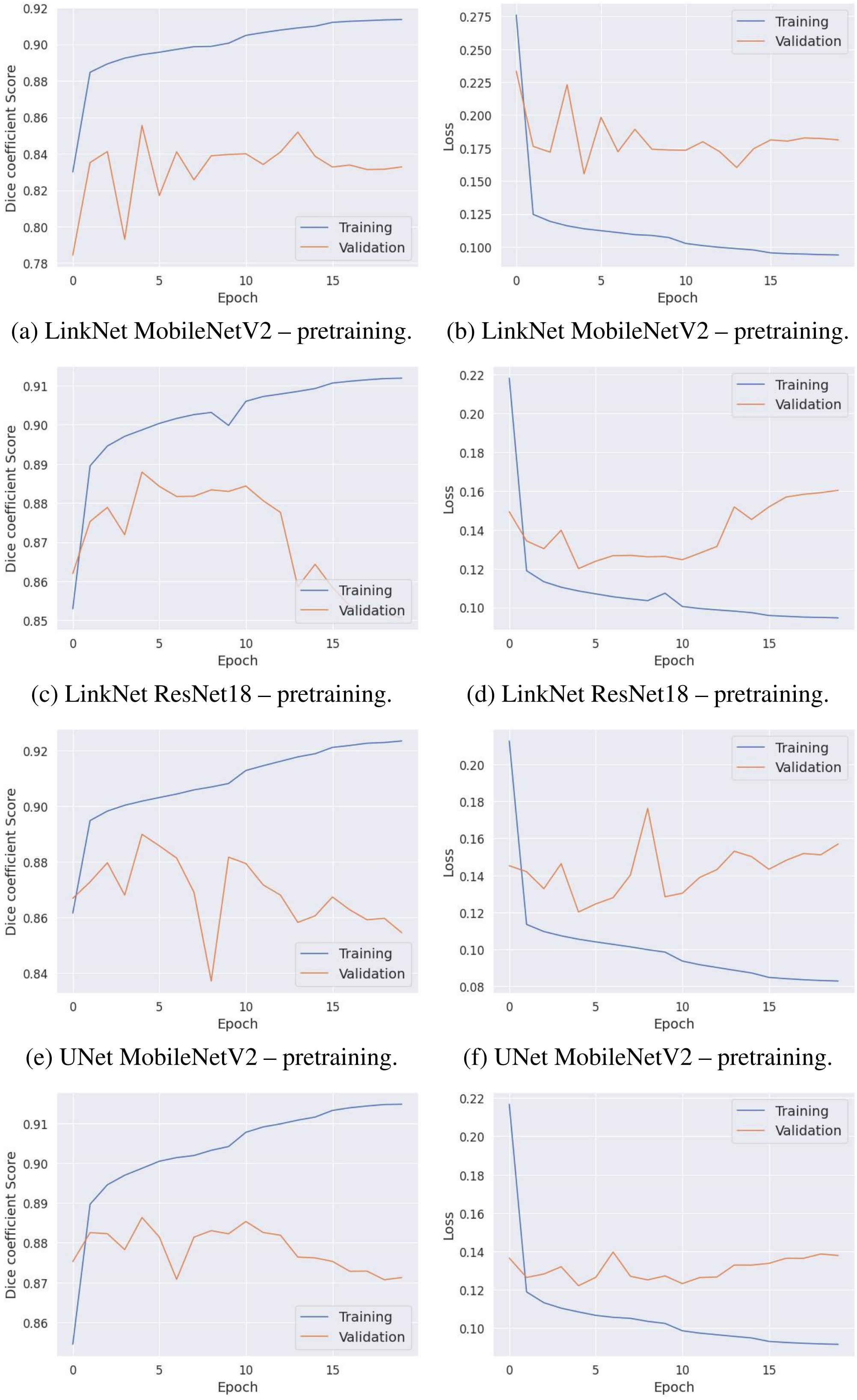
Monzo et al. (2011) applied restrictions to the maximum rotation angle referred to the horizontal in the subject of eye localization: if the pair was rotated more than ±20°, it was discarded. Ahmad et al. (2022) considered that the eyes should be aligned from ±30°. As in-plane face rotations are not usual when using the computer, the palpebral fissure candidates are also usually aligned, and a reasonable threshold can be used to indicate a segmentation fault.

5.3 Pretraining with CelebAMask-HQ samples only

5.3.1 Manual Search

Figure 5.7 shows the Dice coefficient score and the loss for the best models, in terms of validation Dice, found by the manual search, with validation combined loss as the objective being optimized.

Figure 5.7: Dice score and loss – Manual Search pretraining with CelebAMask-HQ



(a) LinkNet MobileNetV2 – pretraining. (b) LinkNet MobileNetV2 – pretraining.

(c) LinkNet ResNet18 – pretraining. (d) LinkNet ResNet18 – pretraining.

(e) UNet MobileNetV2 – pretraining. (f) UNet MobileNetV2 – pretraining.

(g) UNet ResNet18 – pretraining. (h) UNet ResNet18 – pretraining.

Left: Dice Coefficient Score. Right: Loss function.

Source: Author.

All models soon achieved a high validation Dice coefficient score. This is due to the pretrained backbones, and probably is also an effect of the use of batch normalization. They also have a noticeable overfit. It should be noted, however, that the models are not yet fine-tuned. Still, one can observe a better performance in this metric for the UNet models and the LinkNet model with ResNet18 backbone. This also holds true when looking to the loss of the models.

The Table 5.1 shows the confusion matrices for the models obtained with the manual search of learning rate 0.0015 and regularization factor 0.005, considering that, if one eye is not fully closed, eyes are not fully closed. As already mentioned, this test is rather strict, in the sense that even a 1 pixel wrong in a segmentation mask that should be empty makes it a false negative, instead of a true positive.

Table 5.1: Confusion matrices (one eye not fully closed, eyes not fully closed) for the pretrained models in manual search

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5814	3
		5815	2
		5815	2
	Positive: both eyes closed	32	9
		38	3
		38	3

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5815	2
		5815	2
		5814	3
	Positive: both eyes closed	34	7
		38	3
		35	6

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5817	0
		5814	3
		5815	2
	Positive: both eyes closed	41	0
		36	5
		39	2

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5816	1
		5816	1
		5815	2
	Positive: both eyes closed	39	2
		40	1
		37	4

(d) UNet ResNet18.

Considers that, if at least one eye is not fully closed, eyes are not fully closed.

Source: Author.

The Table 5.2 shows the confusion matrices for the models obtained with the manual search of learning rate 0.0015 and regularization factor 0.005, considering that, if at least one eye is fully closed, eyes are fully closed.

Table 5.2: Confusion matrices (one eye fully closed, eyes fully closed) for the pretrained models in manual search

		Predicted label	
		Negative	Positive
	Negative:	5768	49
True	both eyes	5774	43
label	open	5779	38
	Positive: at	22	19
	least 1 eye	25	16
	closed	22	19

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5774	43
True	both eyes	5782	35
label	open	5779	38
	Positive: at	18	23
	least 1 eye	27	14
	closed	22	19

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
	Negative:	5800	17
True	both eyes	5769	48
label	open	5783	34
	Positive: at	32	9
	least 1 eye	24	17
	closed	24	17

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5795	22
True	both eyes	5783	34
label	open	5767	50
	Positive: at	21	20
	least 1 eye	30	11
	closed	22	19

(d) UNet ResNet18.

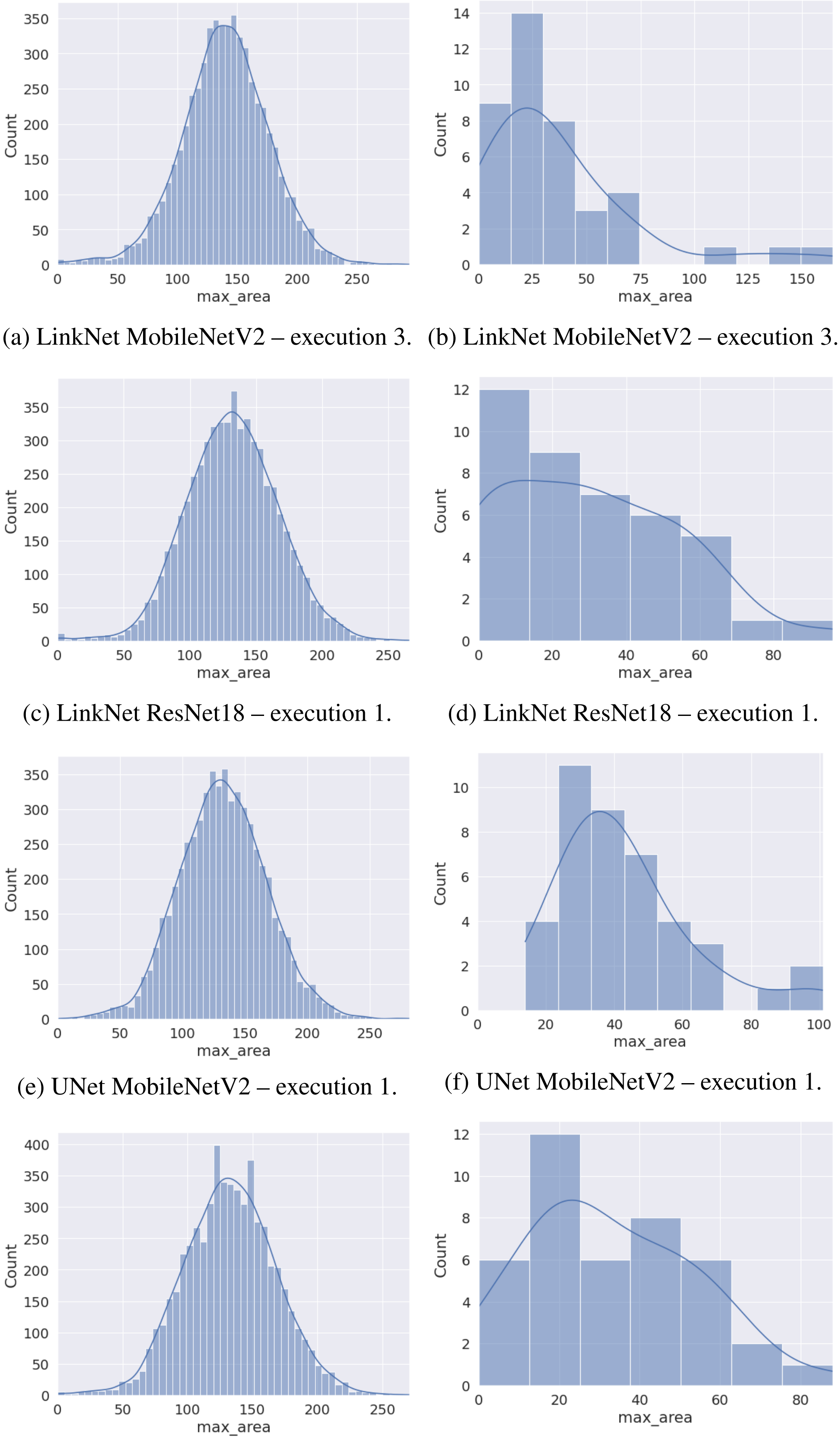
Considers that, if at least one eye is fully closed, eyes are fully closed.

Source: Author.

This consideration, that eyes normally close together in a spontaneous blink, has increased the perception of all models that both eyes are closed when they were annotated as closed eyes, which translates to an increase in true positives and a decrease in false positives. Nevertheless, this also has a negative impact on the eyes considered open that are truly open: there was an increase in false negatives. Fogelton and Benesova (2018) evaluated each eye separately because their completeness can differ, but they noted that if only one of the eye blink detection is enough to report blink, this can dramatically change the results.

The Figure 5.8 shows the maximal area histogram for the test set (and the subset of CelebAMask-HQ with only closed eyes) for the pretrained models in manual search for the best execution in terms of Dice score value. All models returned a bell-shaped histogram for the test set. The maximal area histogram for closed eyes test set indicates that sometimes, the models classified only some pixels incorrectly. Interestingly, UNet MobileNetV2 best model always predicts that there are at least a part of a palpebral fissure open (or a connected-component that is taken by as part of palpebral fissure), probably because of class imbalance.

Figure 5.8: Maximal area histogram – Manual Search with CelebAMask-HQ (pretraining)



(a) LinkNet MobileNetV2 – execution 3. (b) LinkNet MobileNetV2 – execution 3.

(c) LinkNet ResNet18 – execution 1. (d) LinkNet ResNet18 – execution 1.

(e) UNet MobileNetV2 – execution 1. (f) UNet MobileNetV2 – execution 1.

(g) UNet ResNet18 – execution 1. (h) UNet ResNet18 – execution 1.

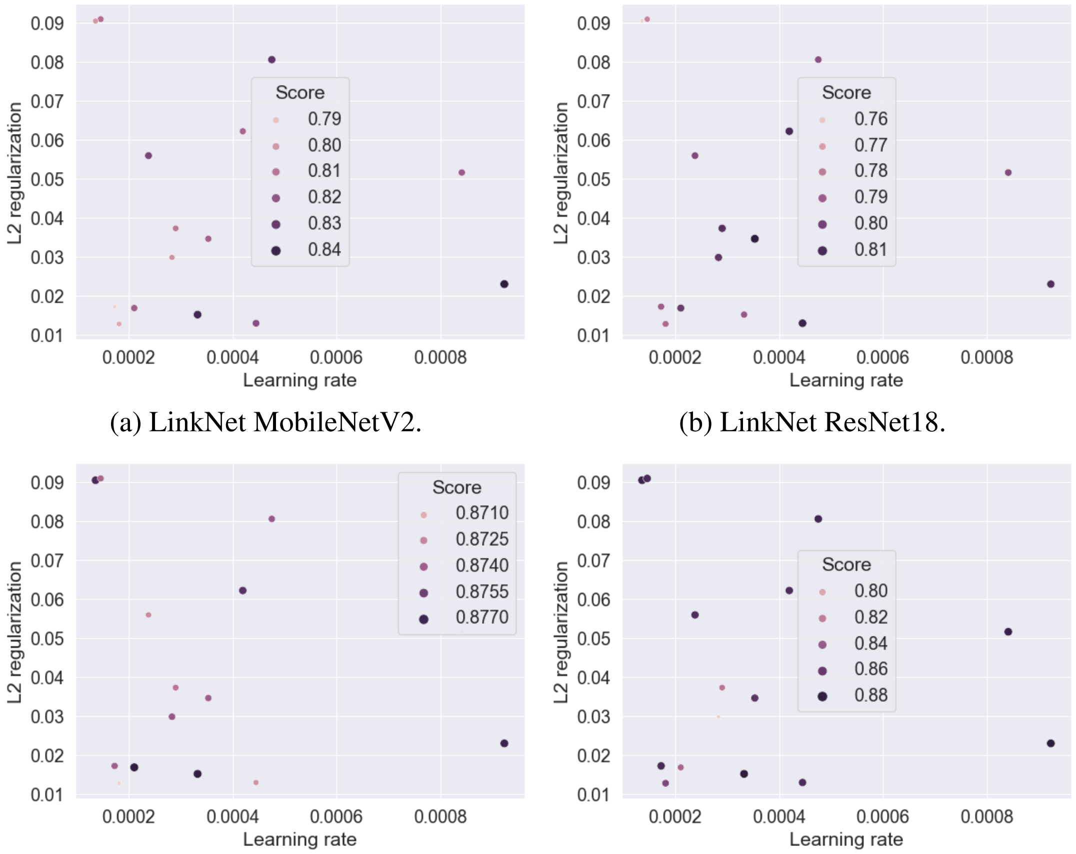
Left: Complete test set. Right: Closed eyes test set (CelebAMask-HQ subset).

Source: Author.

5.3.2 Random Search – 15 trials, 1 execution per trial

The Figure 5.9 shows the hyperparameters search space using Random Search for the different models and topologies. As it can be seen, the results for UNet tend to be superior to those of LinkNet models, with UNet MobileNetV2 having a similar performance for the various pairs combinations of the selected hyperparameters.

Figure 5.9: Hyperparameters search space (Random Search): 15 trials with validation Dice score as objective



(a) LinkNet MobileNetV2.

(b) LinkNet ResNet18.

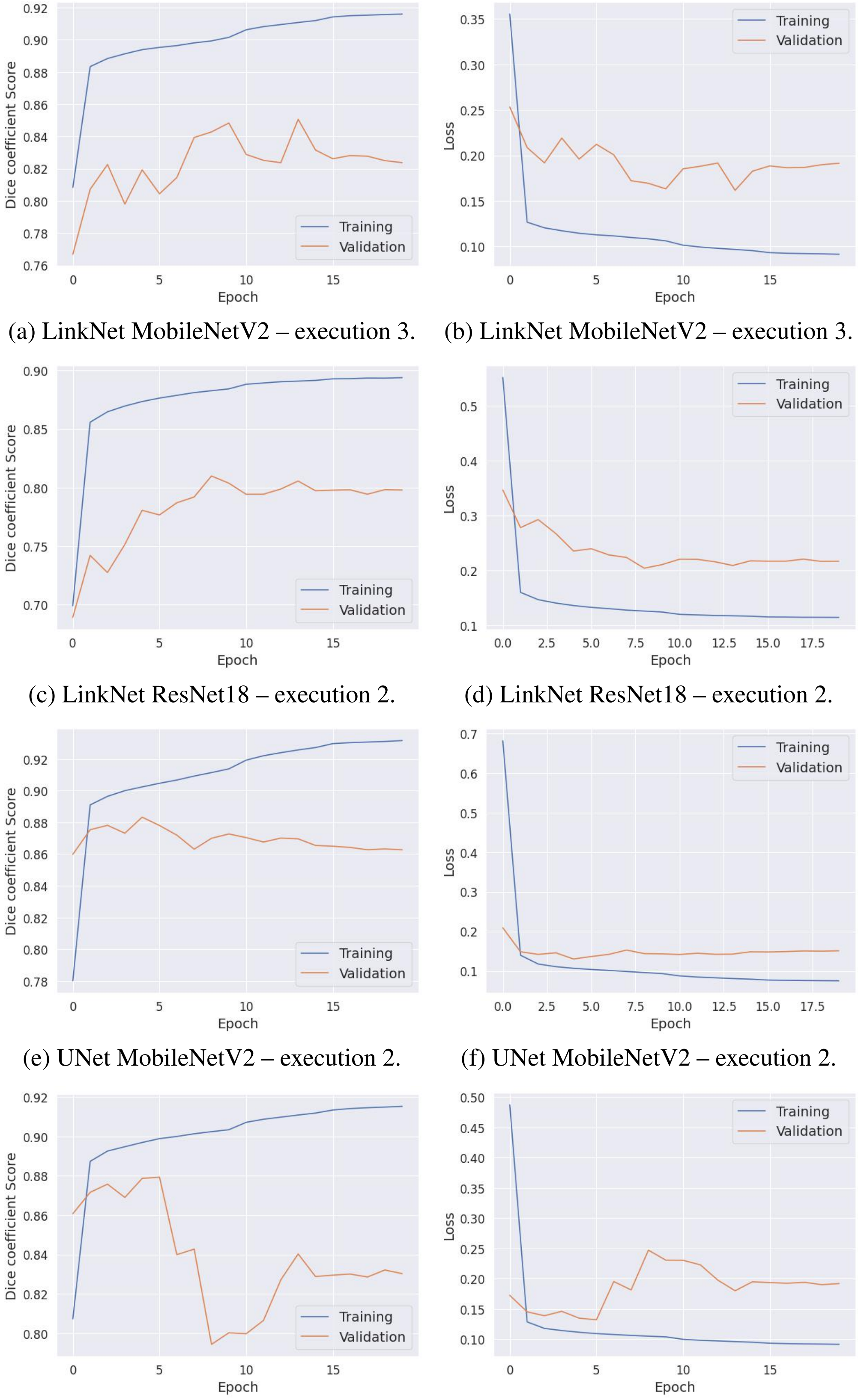
(c) UNet MobileNetV2.

(d) UNet ResNet18.

Source: Author.

The Figure 5.10 shows the Dice score and the loss for the best model found by the Random Search in the 15 trials per model, with validation Dice score as objective. All models display some overfit, as the Dice coefficient score for the training set is greater than 0.90 but that is not the case for the validation set and the distance between the training and the validation curves increases over the epochs. It should be noted, however, that the models are not yet fine-tuned. Still, one can observe a better performance in this metric for the UNet models. This also holds true when looking at the loss of the models.

Figure 5.10: Dice score and loss – Random Search pretraining



(a) LinkNet MobileNetV2 – execution 3.

(b) LinkNet MobileNetV2 – execution 3.

(c) LinkNet ResNet18 – execution 2.

(d) LinkNet ResNet18 – execution 2.

(e) UNet MobileNetV2 – execution 2.

(f) UNet MobileNetV2 – execution 2.

(g) UNet ResNet18 – execution 2.

(h) UNet ResNet18 – execution 2.

Left: Dice Coefficient Score. Right: Loss function.

Source: Author.

The Table 5.3 shows the confusion matrices for the models obtained with random search with 15 trials, 1 execution per trial, considering that, if one eye is not fully closed, eyes are not fully closed.

Table 5.3: Confusion matrices (one eye not fully closed, eyes not fully closed) for the pretrained models in random search with 15 trials, 1 execution per trial

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5816	1
		5813	4
		5814	3
	Positive: both eyes closed	38	3
		37	4
		33	8

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5808	9
		5814	3
		5811	6
	Positive: both eyes closed	33	8
		36	5
		37	4

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5815	2
		5813	4
		5816	1
	Positive: both eyes closed	39	2
		34	7
		39	2

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5816	1
		5815	2
		5815	2
	Positive: both eyes closed	41	0
		39	2
		38	3

(d) UNet ResNet18.

Considers that, if at least one eye is not fully closed, eyes are not fully closed.

Source: Author.

The Table 5.4 shows the confusion matrices for the models obtained with random search with 15 trials, 1 execution per trial, considering that, if at least one eye is fully closed, eyes are fully closed.

Table 5.4: Confusion matrices (one eye fully closed, eyes fully closed) for the pretrained models in random search with 15 trials, 1 execution per trial

		Predicted label	
		Negative	Positive
	Negative:	5784	33
True	both eyes	5773	44
label	open	5754	63
	Positive: at	23	18
	least 1 eye	22	19
	closed	15	26

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5748	69
True	both eyes	5736	81
label	open	5768	49
	Positive: at	17	24
	least 1 eye	23	18
	closed	21	20

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
	Negative:	5795	22
True	both eyes	5762	55
label	open	5792	25
	Positive: at	25	16
	least 1 eye	14	27
	closed	26	15

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5781	36
True	both eyes	5777	40
label	open	5778	39
	Positive: at	27	14
	least 1 eye	25	16
	closed	27	14

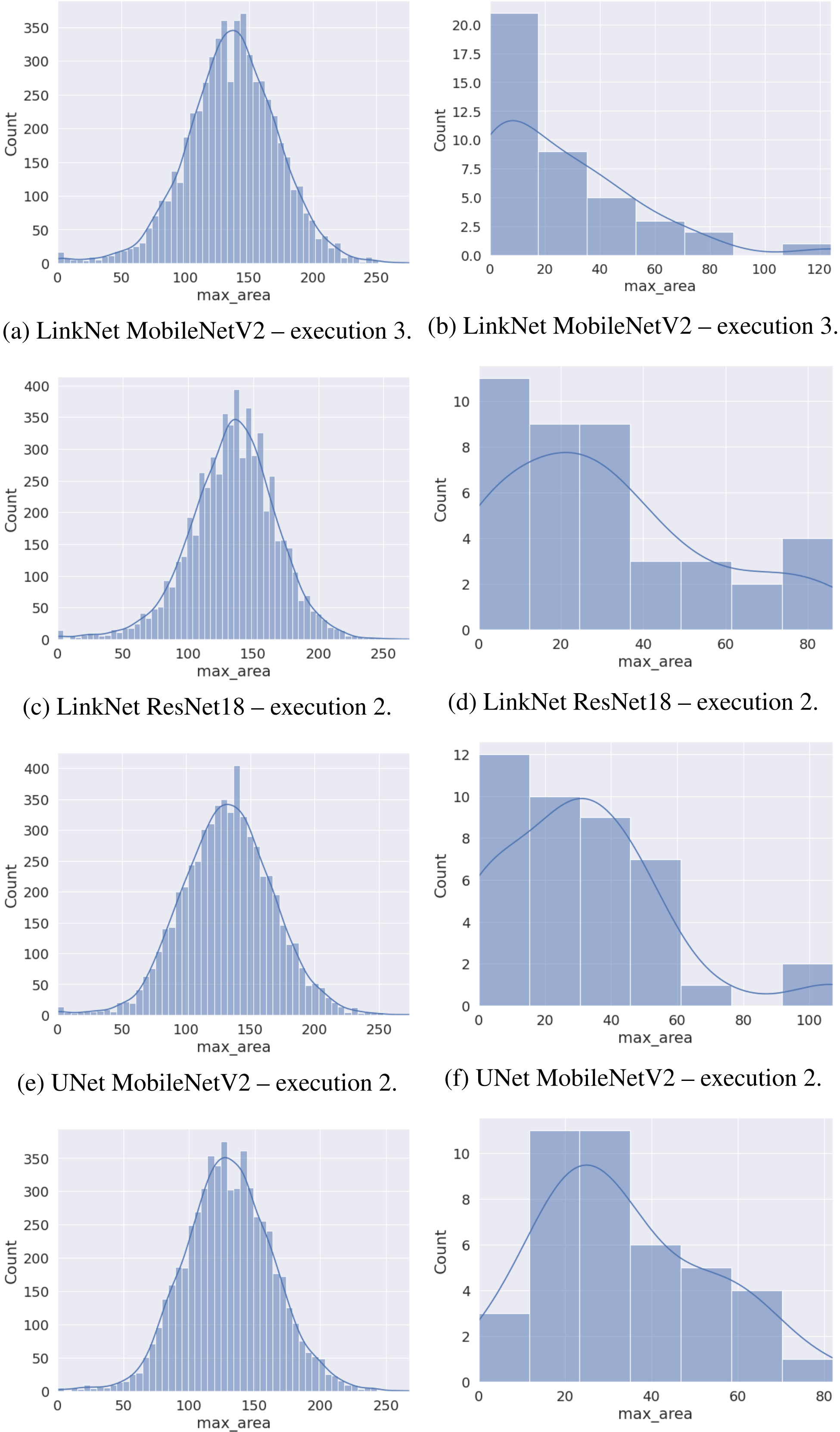
(d) UNet ResNet18.

Considers that, if at least one eye is fully closed, eyes are fully closed.

Source: Author.

The Figure 5.11 shows the maximal area histogram for the open eyes test set for the pretrained models in random search with 15 trials, 1 executions per trial. Again, all models returned a bell-shaped histogram for the test set.

Figure 5.11: Maximal area histogram – Random Search with 15 trials (pretraining)



(a) LinkNet MobileNetV2 – execution 3.

(b) LinkNet MobileNetV2 – execution 3.

(c) LinkNet ResNet18 – execution 2.

(d) LinkNet ResNet18 – execution 2.

(e) UNet MobileNetV2 – execution 2.

(f) UNet MobileNetV2 – execution 2.

(g) UNet ResNet18 – execution 2.

(h) UNet ResNet18 – execution 2.

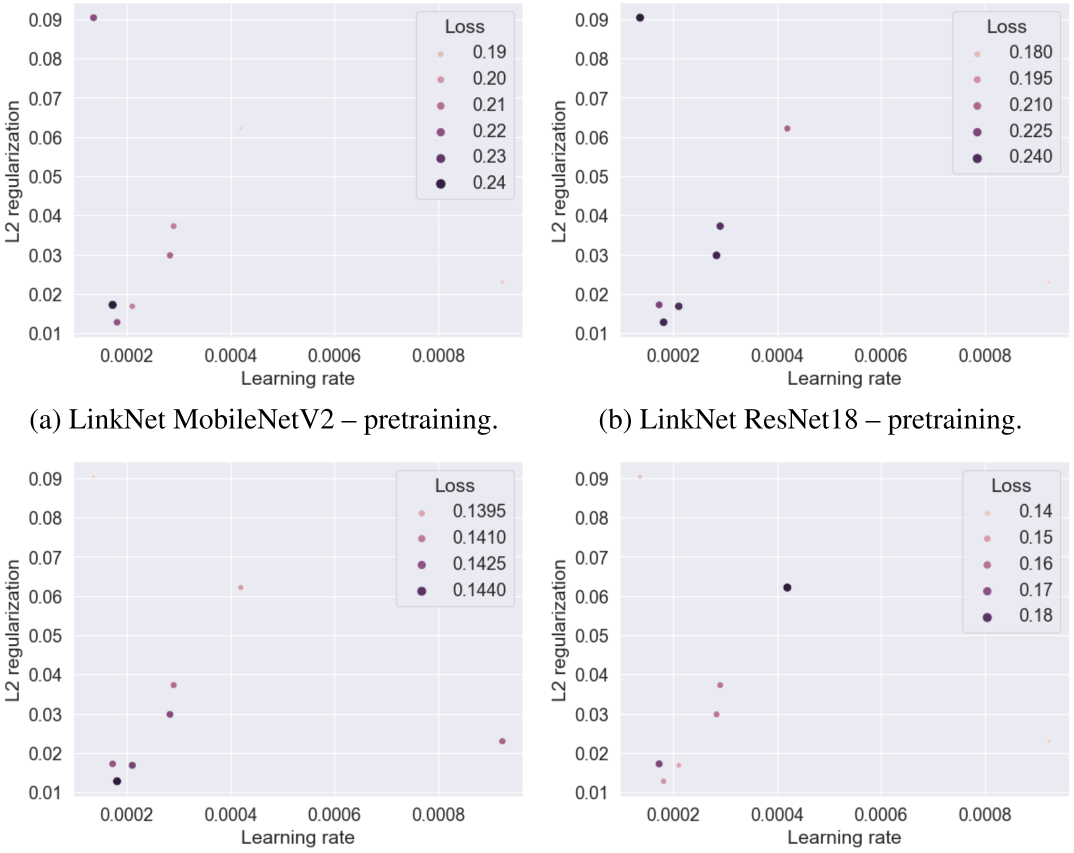
Left: Complete test set. Right: Closed eyes test set (subtest of CelebAMask-HQ).

Source: Author.

5.3.3 Random Search – 8 trials, 3 executions per trial

The Figure 5.12 shows the hyperparameters search space using Random Search for the different models and topologies. Again, the results for UNet tend to be superior to those of LinkNet models, with UNet MobileNetV2 having a similar performance for the various pairs combinations of the selected hyperparameters.

Figure 5.12: Hyperparameters search space (Random Search): 8 trials, 3 executions per trial, with validation loss as objective



(a) LinkNet MobileNetV2 – pretraining.

(b) LinkNet ResNet18 – pretraining.

(c) UNet MobileNetV2 – pretraining.

(d) UNet ResNet18 – pretraining.

Source: Author.

Table 5.5 shows the confusion matrices for the models obtained with random search with 8 trials, 3 execution per trial, considering that, if one eye is not fully closed, eyes are not fully closed.

Table 5.5: Confusion matrices (one eye not fully closed, eyes not fully closed) for the pretrained models in random search with 8 trials, 3 executions per trial

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5782	35
	Positive: both eyes closed	15	26

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5799	18
	Positive: both eyes closed	15	26

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5813	4
	Positive: both eyes closed	28	13

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5798	19
	Positive: both eyes closed	14	27

(d) UNet ResNet18.

Considers that, if at least one eye is not fully closed, eyes are not fully closed.

Source: Author.

Comparing the Table to Table 5.3 and Table 5.1, there is a noticeable reduction in incorrect predictions indicating that the segmentation mask is empty, with an increase for the true positives. Models with ResNet18 have a better balance between false positive and false negative with the selected hyperparameters and the combined loss, that was also used in Manual Search.

Table 5.6 shows the confusion matrices for the models obtained with random search with 8 trials, 3 execution per trial, considering that, if at least one eye is fully closed, eyes are fully closed. Again, there is an increase in the detection of closed eyes, both for true and false positives.

Table 5.6: Confusion matrices (one eye fully closed, eyes fully closed) for the pretrained models in random search with 8 trials, 3 executions per trial

		Predicted label	
		Negative	Positive
True label	Negative: both eyes open	5618	199
	Positive: at least 1 eye closed	3	38

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: both eyes open	5693	124
	Positive: at least 1 eye closed	3	38

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
True label	Negative: both eyes open	5771	46
	Positive: at least 1 eye closed	14	27

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: both eyes open	5702	115
	Positive: at least 1 eye closed	3	38

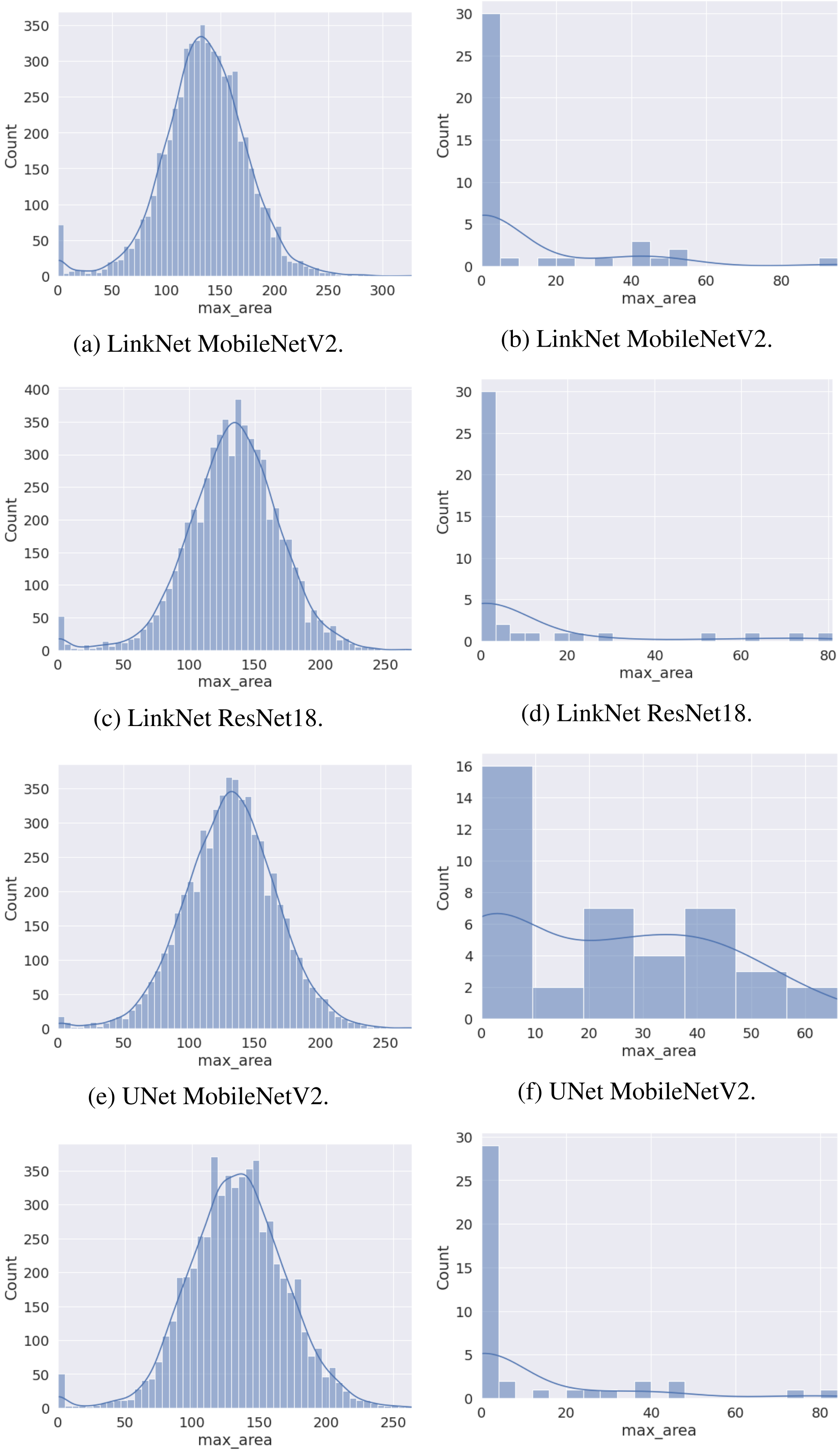
(d) UNet ResNet18.

Considers that, if at least one eye is fully closed, eyes are fully closed.

Source: Author.

The Figure 5.13 shows Maximal area histogram for open eyes test set for the pretrained models in random search with 8 trials, 3 executions per trial. All models present a bell-shaped histogram for the test set, but there is a noticeable peak close to 0 for LinkNet models and the UNet ResNet18.

Figure 5.13: Maximal area histogram – Random Search with 24 executions (pretraining)



(a) LinkNet MobileNetV2.

(b) LinkNet MobileNetV2.

(c) LinkNet ResNet18.

(d) LinkNet ResNet18.

(e) UNet MobileNetV2.

(f) UNet MobileNetV2.

(g) UNet ResNet18.

(h) UNet ResNet18.

Left: Test set. Right: Closed eyes test set (CelebAMask-HQ subset). 8 trials, 3 executions/trial.

Source: Author.

5.3.4 Results

Appendix D Tables of results of experiments contains the experiment's outcomes in detail. The Table D.1 displays the results for the experiments with CelebAMask-HQ dataset, in descending order for the Dice score. The Table D.2 summarizes metrics values, while the Table D.3 displays the inference time experiments results.

The best model in terms of Dice Score for the whole test set was a UNet MobileNetV2 (0.8885), followed by LinkNet ResNet18 (0.8878) with learning rate 0.0015 and $\lambda = 0.005$. Considering these choice of hyperparameters, the mean ± standard deviation Dice Score of these models for the 3 trials was:

- UNet MobileNetV2: 0.8874 ± 0.0014;
- LinkNet ResNet18: $0,8857 \pm 0.0023$.

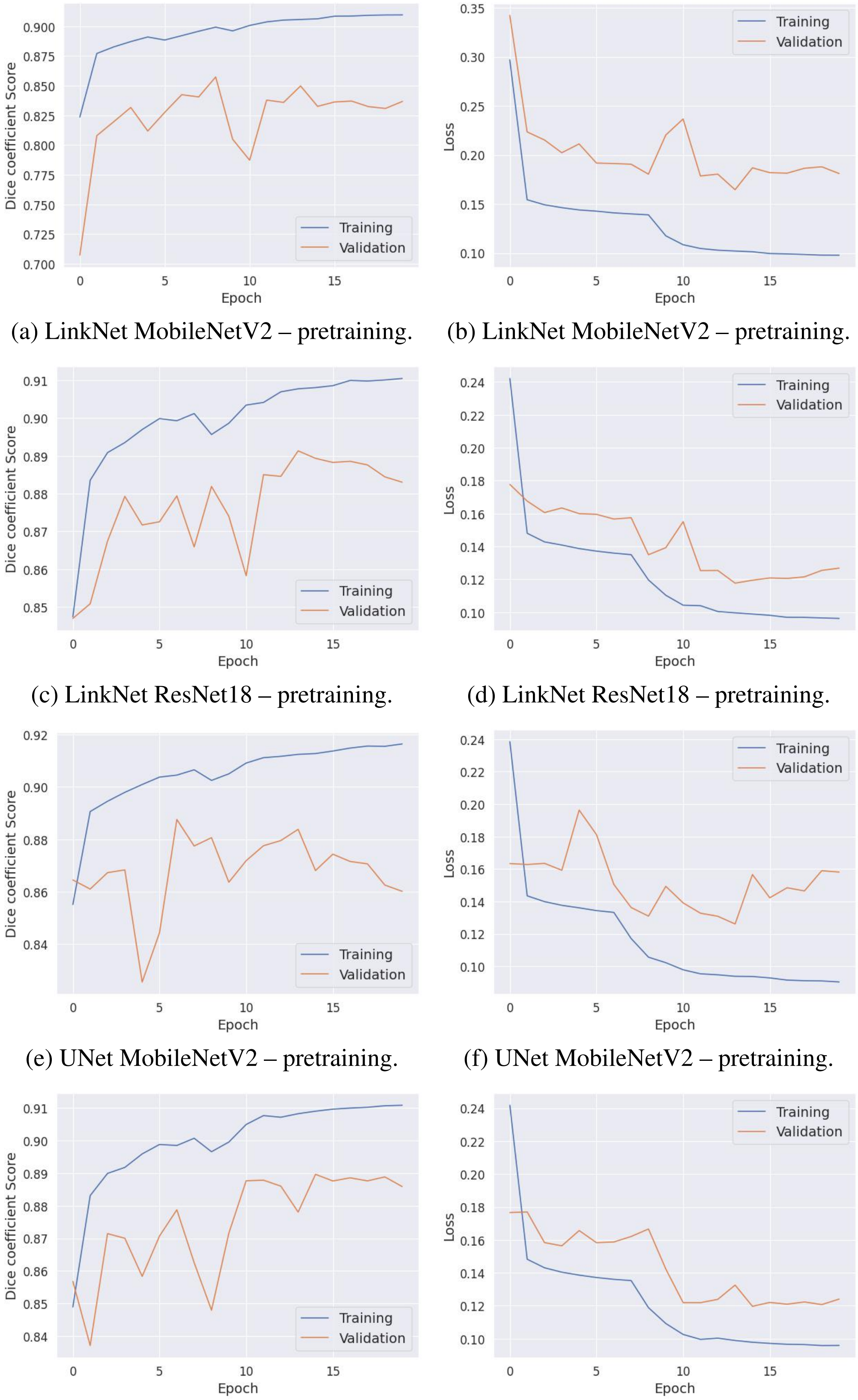
Interestingly, the worst performance was also one of a LinkNet ResNet18 model (0.7936), with a mean ± standard deviation of these models equal to $0,8008 \pm 0.0080$ for learning rate 0.000353 and $\lambda = 0.0346$, indicating the importance of adjusting these hyperparameters in performance. UNet's models, specially MobileNetV2 ones, on the other hand, had a generally better performance independently of the choice of hyperparameters.

Concerning the time inferences shown in Table D.3 displays, it can be observed that there is a time difference between using predicting a single image or a batch. In fact, it is not possible to predict the images in real-time as the frames are retrieved. However, these times drop considerably if a batch of images is given to the GPU. There seems to be a limit in performance for predict single inputs that may arise from the CPU to GPU communication, as for the same model/backbone, predicting 30 batch is faster in Tesla V100 GPU than Tesla T4, but that is not the case for a single image. Inference times for a single image are also similar when using *predict* method between architectures e backbones. It is also interesting to observe the *call* and *predict* do not have a similar performance only for MobileNetV2 backbones, regardless the architecture.

5.4 Pretraining adding CEW closed eyes samples

The Figure 5.14 shows the Dice coefficient score and the loss for the best models, in terms of validation Dice, found by the Manual Search, with validation combined loss as the objective being optimized.

Figure 5.14: Dice score and loss – pretraining (CelebAMask-HQ and CEW)



(a) LinkNet MobileNetV2 – pretraining. (b) LinkNet MobileNetV2 – pretraining.

(c) LinkNet ResNet18 – pretraining. (d) LinkNet ResNet18 – pretraining.

(e) UNet MobileNetV2 – pretraining. (f) UNet MobileNetV2 – pretraining.

(g) UNet ResNet18 – pretraining. (h) UNet ResNet18 – pretraining.

Left: Dice Coefficient Score. Right: Loss function.

Source: Author.

All models soon achieved a high validation Dice coefficient score, again due to the pretrained backbones, and probably is also an effect of the use of batch normalization. There seems to be a trend that the best epoch of the models (the one kept by early stopping saving technique) occurs later than in previous experiments. They also still have a noticeable overfit, but is specially visible that the models with ResNet18 backbone, were benefited by the additional images. It should be noted, however, that the models are not yet fine-tuned. As in the previous manual search, one can observe a better performance of the validation Dice metric for the UNet models and the LinkNet model with ResNet18 backbone. This also holds true here and when looking at the loss.

Comparing the validation loss curves of Figure 5.7 and Figure 5.14 for ResNet18 backbone models show the reduction of overfit (or, at least, that early stopping happens latter).

The Table 5.7 shows the confusion matrices for the models obtained with the manual search of learning rate 0.0015 and regularization factor 0.005, considering that, if one eye is not fully closed, eyes are not fully closed.

Table 5.7: Confusion matrices (one eye not fully closed, eyes not fully closed) for the pretrained models in manual search with CEW

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5813	4
		5814	3
		5810	7
	Positive: both eyes closed	35	6
		37	4
		31	10

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5799	18
		5805	12
		5793	24
	Positive: both eyes closed	5	36
		13	28
		5	36

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5792	25
		5811	6
		5817	0
	Positive: both eyes closed	13	28
		27	14
		32	9

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5797	20
		5800	17
		5806	11
	Positive: both eyes closed	6	35
		10	31
		8	33

(d) UNet ResNet18.

Considers that, if at least one eye is not fully closed, eyes are not fully closed.

Source: Author.

The Table 5.8 shows the confusion matrices for the models obtained with the manual search of learning rate 0.0015 and regularization factor 0.005, if at least one eye is fully closed, eyes are fully closed.

Table 5.8: Confusion matrices (one eye fully closed, eyes fully closed) for the pretrained models in manual search with CEW

		Predicted label	
		Negative	Positive
	Negative:	5775	42
True	both eyes	5778	39
label	open	5761	56
	Positive: at	20	21
	least 1 eye	18	23
	closed	15	26

(a) LinkNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5779	107
True	both eyes	5730	87
label	open	5707	110
	Positive: at	2	39
	least 1 eye	5	36
	closed	2	39

(b) LinkNet ResNet18.

		Predicted label	
		Negative	Positive
	Negative:	5704	113
True	both eyes	5751	66
label	open	5782	35
	Positive: at	3	38
	least 1 eye	8	33
	closed	20	21

(c) UNet MobileNetV2.

		Predicted label	
		Negative	Positive
	Negative:	5722	95
True	both eyes	5712	105
label	open	5741	76
	Positive: at	3	38
	least 1 eye	3	38
	closed	2	39

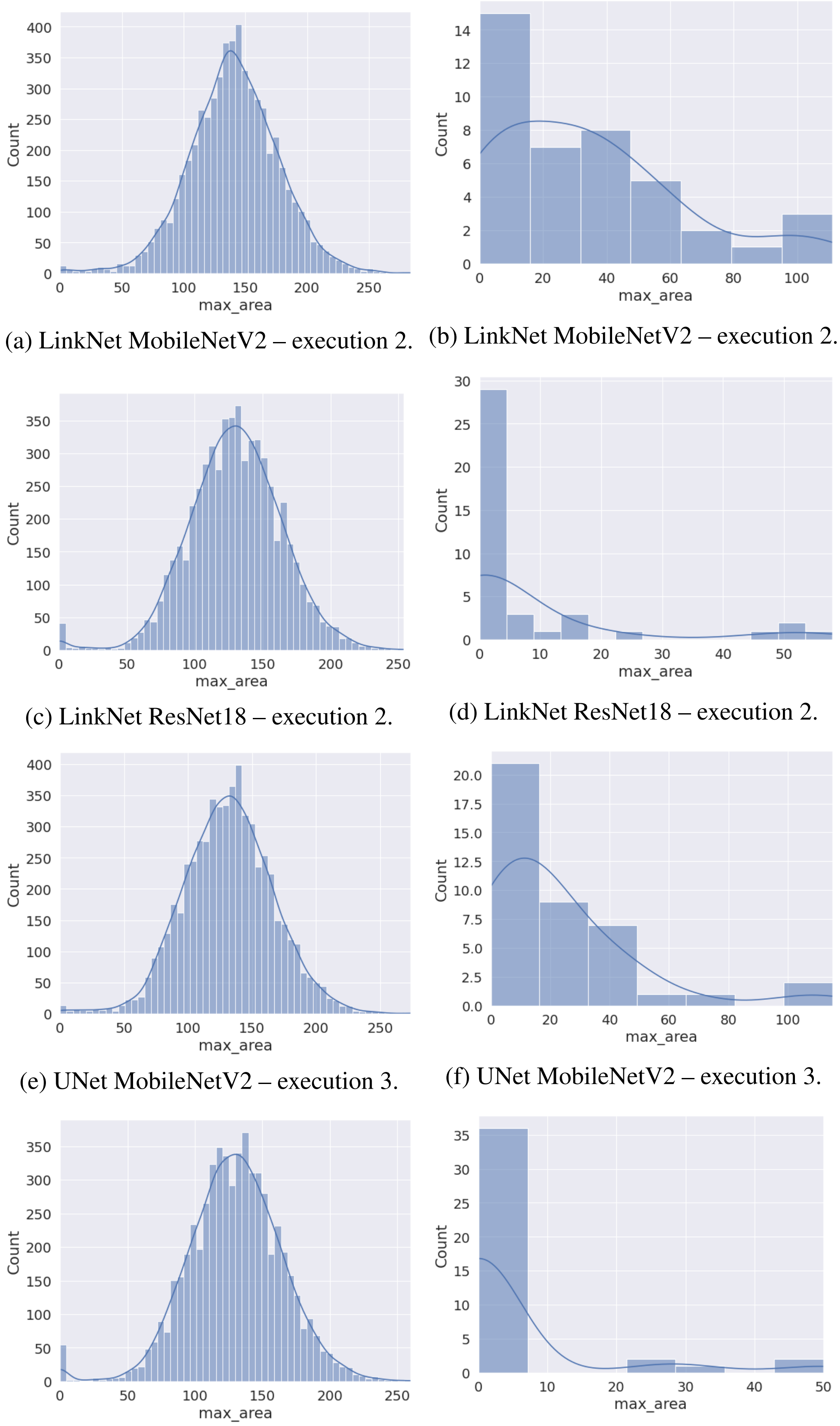
(d) UNet ResNet18.

Considers that, if at least one eye is fully closed, eyes are fully closed.

Source: Author.

The Figure 5.15 shows the maximal area histogram for the test set for the pretrained models in manual search with CEW dataset for the best execution in terms of Dice score value.

Figure 5.15: Maximal area histogram – pretraining (CelebAMask-HQ and CEW)



(a) LinkNet MobileNetV2 – execution 2.

(b) LinkNet MobileNetV2 – execution 2.

(c) LinkNet ResNet18 – execution 2.

(d) LinkNet ResNet18 – execution 2.

(e) UNet MobileNetV2 – execution 3.

(f) UNet MobileNetV2 – execution 3.

(g) UNet ResNet18 – execution 1.

(h) UNet ResNet18 – execution 1.

Left: Complete test set. Right: Closed eyes test set (subtest of CelebAMask-HQ).

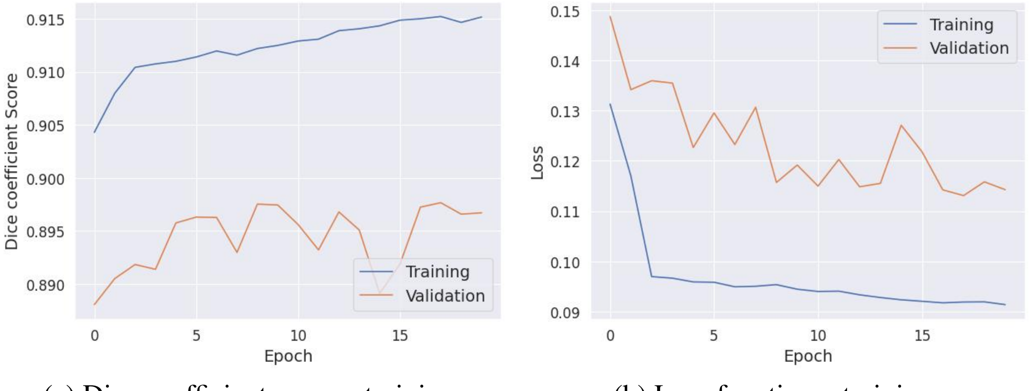
Source: Author.

The Table D.4 displays the results for the experiments with CelebAMask-HQ and CEW for the pretraining. Again, the best model in terms of Dice Score for the whole test set was an UNet MobileNetV2 0.8894, followed by LinkNet ResNet18 (0.8855). This UNet MobileNetV2 model was then chosen to be fine-tuned. It is interesting to point out that even if the performance of this model in terms of Dice metric was a little inferior to the one with only CelebAMask-HQ, it would still make sense to choose this model for training, with both datasets, as it has been trained in more diverse data.

5.5 Training

The Figure 5.16 shows the Dice coefficient score and Loss function for the best UNet MobileNetV2 model in terms of Dice coefficient score, that was obtained in the second execution. As expected, the Dice value is high since the first epoch. The best value was obtained at the epoch 18. The validation loss shows a decreasing behavior with some oscillations.

Figure 5.16: Dice coefficient score and Loss for the best UNet MobileNetV2 model



(a) Dice coefficient score – training.

(b) Loss function – training.

Source: Author.

The Table 5.9 shows the confusion matrices for the UNet MobileNetV2 trained in CelebAMask-HQ and CEW, while the Figure 5.17 shows the Maximal area histogram for the best UNet MobileNetV2 trained model.

Table 5.9: Confusion matrices for the trained UNet MobileNetV2 models

		Predicted label	
		Negative	Positive
True label	Negative: at least 1 eye not closed	5815 **5806** 5811	2 **11** 6
	Positive: both eyes closed	30 **13** 25	11 **28** 16

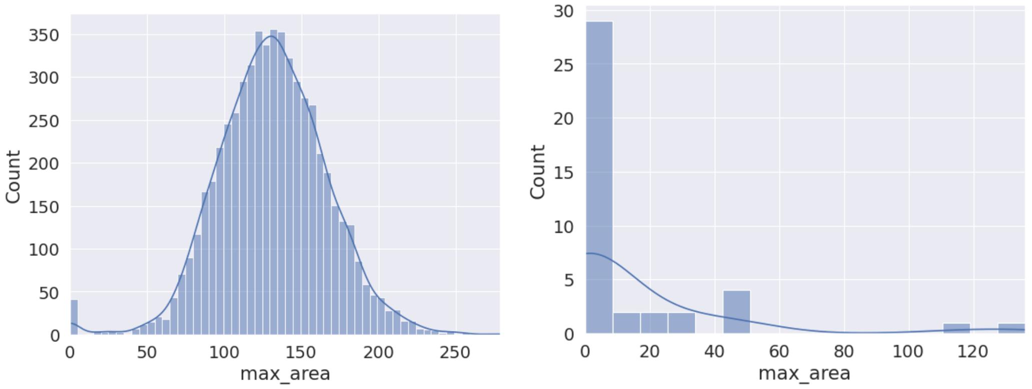
(a) Considers that, if at least one eye is not fully closed, eyes are not fully closed.

		Predicted label	
		Negative	Positive
True label	Negative: both eyes open	5790 **5767** 5778	27 **50** 39
	Positive: at least 1 eye not closed	16 **5** 10	25 **36** 31

(b) Considers that, if at least one eye is fully closed, eyes are fully closed.

Source: Author.

Figure 5.17: Maximal area histogram for the trained models



(a) UNet MobileNetV2: test set – execution 2.

(b) UNet MobileNetV2: closed eyes test – execution 2.

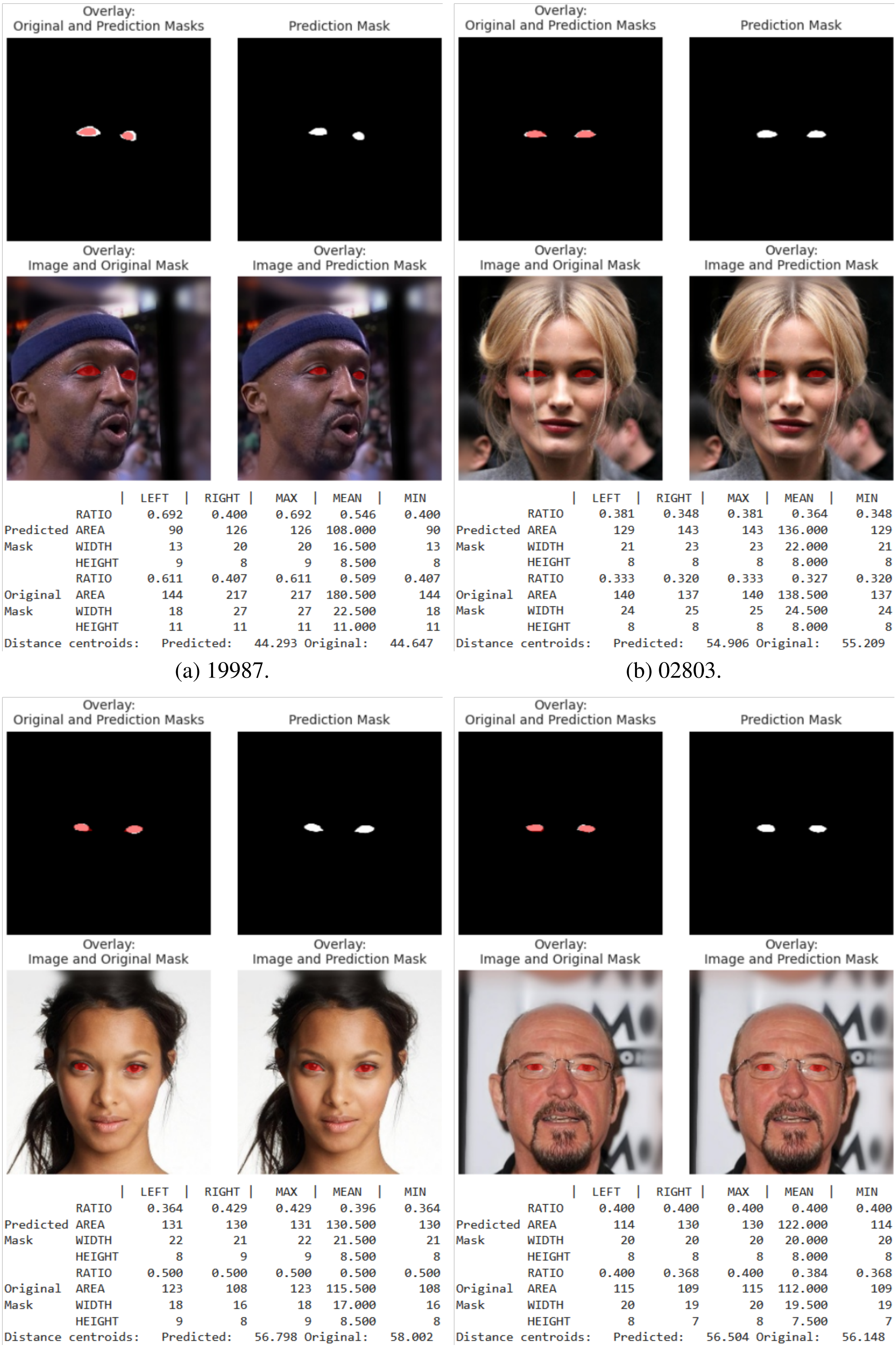
Source: Author.

The Figure 5.18 shows some examples of predicted mask for open eyes. In Figure 5.18a, the annotator selected more than the palpebral fissure, with some part of the eyelid marked. It is interesting to observe that the model did not segment the eyelid, but only the palpebral fissure. In the case of Figure 5.18c, the annotator did not include the corner of the eye, but the model did, completing the palpebral fissure. Figure 5.18b shows a case where both the ground truth and the prediction consider the complete palpebral fissure, regardless of partial hair occlusion. Figure 5.18d exemplifies a prediction for someone using glasses.

Figure 5.19a shows another prediction for someone using glasses. In this case, it is interesting to observe that the prediction of the width of both palpebral fissures was greater than the ground truths, but for the heights it was the contrary. Figure 5.19b shows a case where the ground truth considered the complete palpebral fissure, regardless of partial hair occlusion, but the prediction did not. In Figure 5.19c, the prediction segmentation mask also displays a problem in the segmentation of the right eye. In Figure 5.19d, there is a partial occlusion of the periocular region and the face, and the person is looking down, but the model did not predict any pixel as corresponding to the palpebral fissures.

The Figure 5.20 shows some examples of problematic predicted masks for closed eyes, where eyes were annotated as closed, but the model returned non-empty masks. In Figure 5.20a, the singer's makeup seems to have misled the model. One can observe, however, that the faulty segmentation has a small "interocular distance" and one component is rather small. Furthermore, the angle between the components and the horizontal is a great indication that a flawed segmentation took place. In Figure 5.20b, only one component was predicted in the eye region. In Figure 5.20c and Figure 5.20d, the eyes were annotated as closed, but the model predicted mask in the eyes' region. In the second case, it is a situation that is typically difficult even for humans to indicate the state of the eyes, as the person is squinting them.

Figure 5.18: Examples of predicted masks for open eyes



```
                 |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
          RATIO     0.692    0.400    0.692    0.546    0.400
Predicted AREA         90      126      126  108.000       90
Mask      WIDTH        13       20       20   16.500       13
          HEIGHT        9        8        9    8.500        8
          RATIO     0.611    0.407    0.611    0.509    0.407
Original  AREA        144      217      217  180.500      144
Mask      WIDTH        18       27       27   22.500       18
          HEIGHT       11       11       11   11.000       11
Distance centroids:    Predicted:   44.293 Original:    44.647
```

(a) 19987.

```
                 |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
          RATIO     0.381    0.348    0.381    0.364    0.348
Predicted AREA        129      143      143  136.000      129
Mask      WIDTH        21       23       23   22.000       21
          HEIGHT        8        8        8    8.000        8
          RATIO     0.333    0.320    0.333    0.327    0.320
Original  AREA        140      137      140  138.500      137
Mask      WIDTH        24       25       25   24.500       24
          HEIGHT        8        8        8    8.000        8
Distance centroids:    Predicted:   54.906 Original:    55.209
```

(b) 02803.

```
                 |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
          RATIO     0.364    0.429    0.429    0.396    0.364
Predicted AREA        131      130      131  130.500      130
Mask      WIDTH        22       21       22   21.500       21
          HEIGHT        8        9        9    8.500        8
          RATIO     0.500    0.500    0.500    0.500    0.500
Original  AREA        123      108      123  115.500      108
Mask      WIDTH        18       16       18   17.000       16
          HEIGHT        9        8        9    8.500        8
Distance centroids:    Predicted:   56.798 Original:    58.002
```

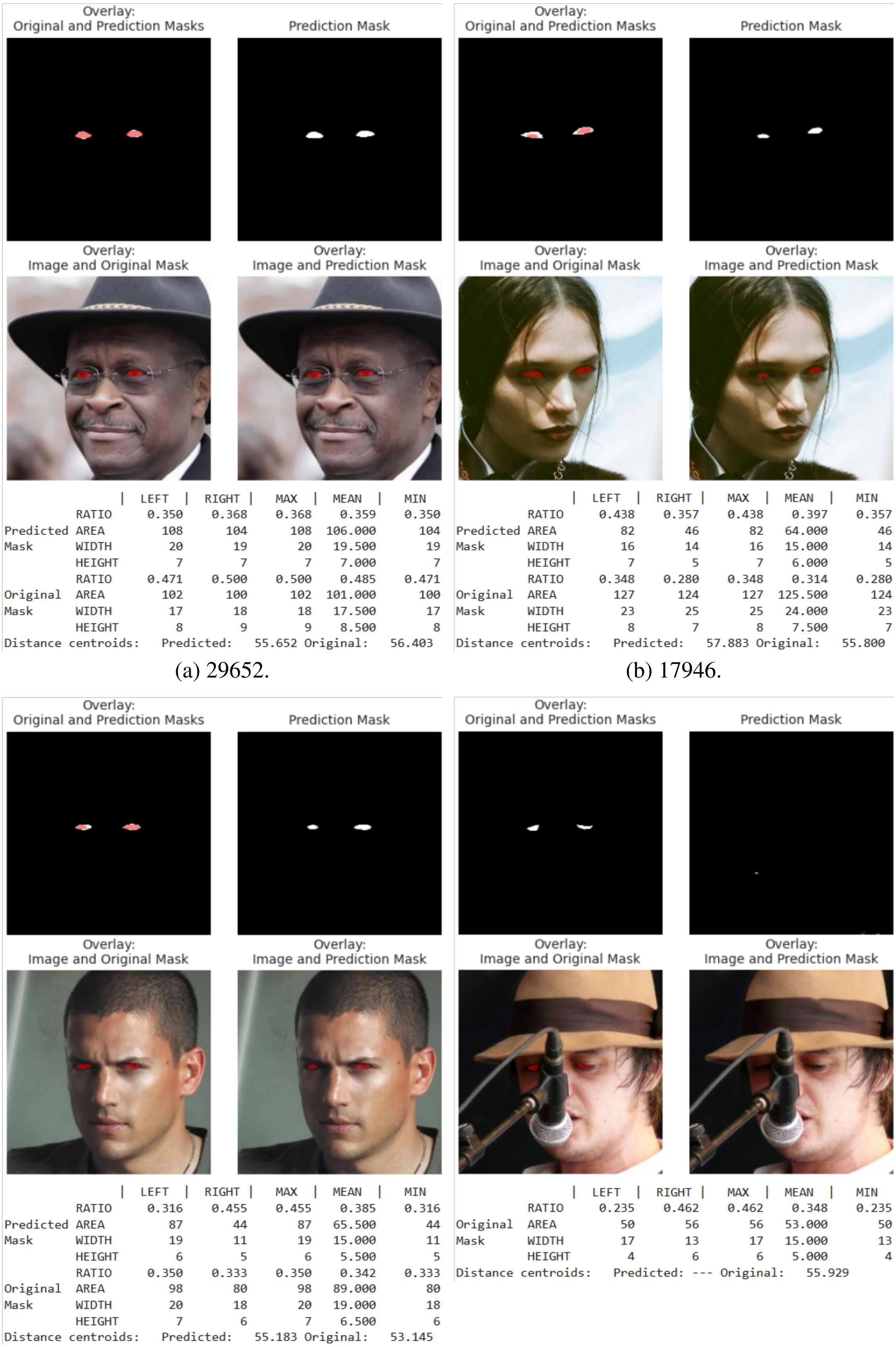
(c) 16858.

```
                 |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
          RATIO     0.400    0.400    0.400    0.400    0.400
Predicted AREA        114      130      130  122.000      114
Mask      WIDTH        20       20       20   20.000       20
          HEIGHT        8        8        8    8.000        8
          RATIO     0.400    0.368    0.400    0.384    0.368
Original  AREA        115      109      115  112.000      109
Mask      WIDTH        20       19       20   19.500       19
          HEIGHT        8        7        8    7.500        7
Distance centroids:    Predicted:   56.504 Original:    56.148
```

(d) 24016.

Source: Author.

Figure 5.19: Examples of predicted masks for open eyes (continuation)



```
               |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
         RATIO    0.350    0.368    0.368    0.359    0.350
Predicted AREA      108      104      108  106.000      104
Mask     WIDTH       20       19       20   19.500       19
         HEIGHT       7        7        7    7.000        7
         RATIO    0.471    0.500    0.500    0.485    0.471
Original AREA       102      100      102  101.000      100
Mask     WIDTH       17       18       18   17.500       17
         HEIGHT       8        9        9    8.500        8
Distance centroids:   Predicted:   55.652 Original:   56.403
```

(a) 29652.

```
               |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
         RATIO    0.438    0.357    0.438    0.397    0.357
Predicted AREA       82       46       82   64.000       46
Mask     WIDTH       16       14       16   15.000       14
         HEIGHT       7        5        7    6.000        5
         RATIO    0.348    0.280    0.348    0.314    0.280
Original AREA       127      124      127  125.500      124
Mask     WIDTH       23       25       25   24.000       23
         HEIGHT       8        7        8    7.500        7
Distance centroids:   Predicted:   57.883 Original:   55.800
```

(b) 17946.

```
               |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
         RATIO    0.316    0.455    0.455    0.385    0.316
Predicted AREA       87       44       87   65.500       44
Mask     WIDTH       19       11       19   15.000       11
         HEIGHT       6        5        6    5.500        5
         RATIO    0.350    0.333    0.350    0.342    0.333
Original AREA        98       80       98   89.000       80
Mask     WIDTH       20       18       20   19.000       18
         HEIGHT       7        6        7    6.500        6
Distance centroids:   Predicted:   55.183 Original:   53.145
```

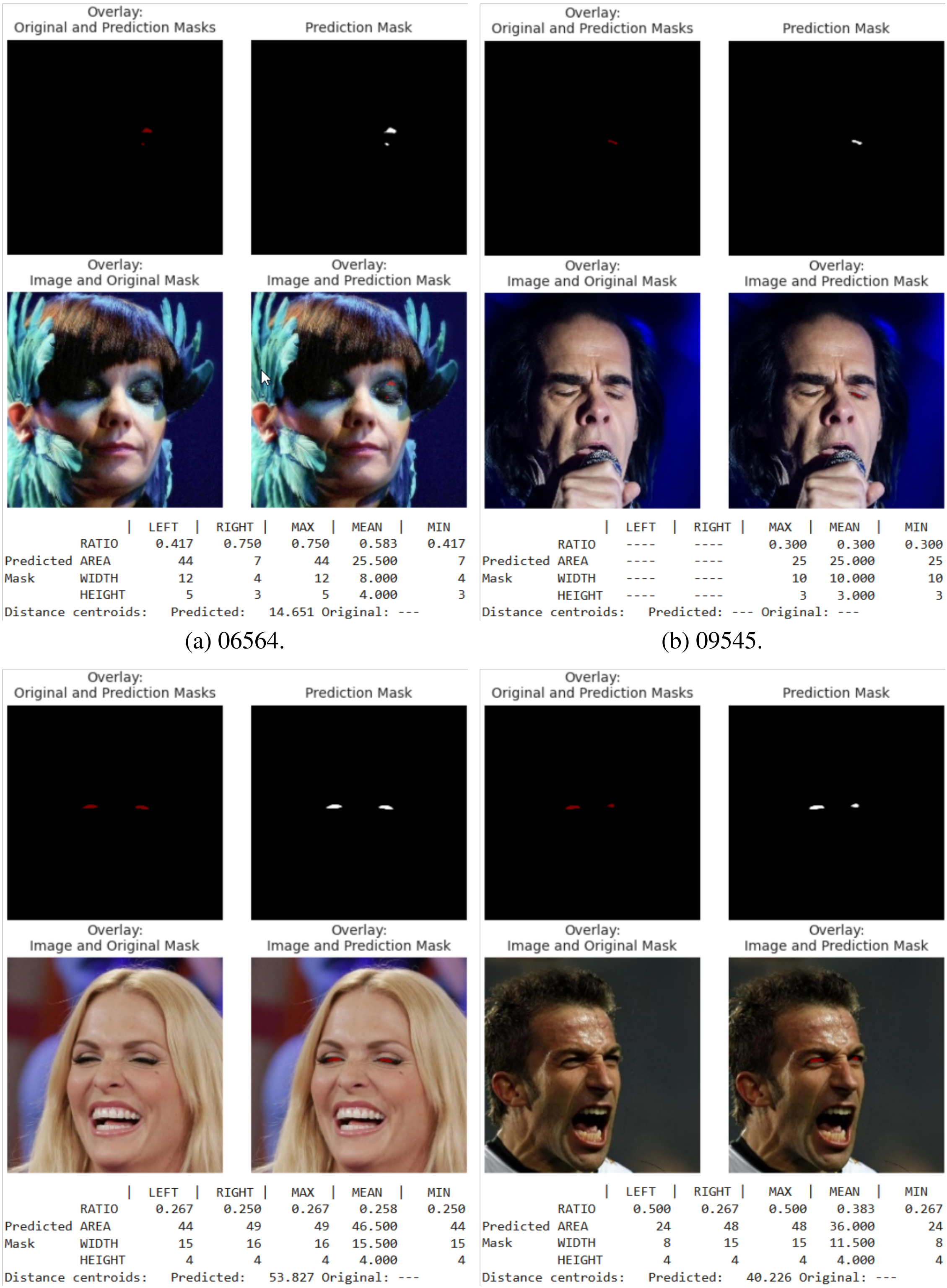
(c) 10114.

```
               |  LEFT  |  RIGHT |   MAX  |   MEAN  |   MIN
         RATIO    0.235    0.462    0.462    0.348    0.235
Original AREA        50       56       56   53.000       50
Mask     WIDTH       17       13       17   15.000       13
         HEIGHT       4        6        6    5.000        4
Distance centroids:   Predicted: --- Original:   55.929
```

(d) 04620.

Source: Author.

Figure 5.20: Examples of problematic predicted masks for closed eyes



```
               |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN
          RATIO   0.417    0.750    0.750   0.583   0.417
Predicted AREA       44        7       44  25.500       7
Mask      WIDTH      12        4       12   8.000       4
          HEIGHT      5        3        5   4.000       3
Distance centroids:   Predicted:   14.651 Original: ---
```

(a) 06564.

```
               |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN
          RATIO   ----     ----     0.300   0.300   0.300
Predicted AREA    ----     ----        25  25.000      25
Mask      WIDTH   ----     ----        10  10.000      10
          HEIGHT  ----     ----         3   3.000       3
Distance centroids:   Predicted: --- Original: ---
```

(b) 09545.

```
               |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN
          RATIO   0.267    0.250    0.267   0.258   0.250
Predicted AREA       44       49       49  46.500      44
Mask      WIDTH      15       16       16  15.500      15
          HEIGHT      4        4        4   4.000       4
Distance centroids:   Predicted:   53.827 Original: ---
```

(c) 13200.

```
               |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN
          RATIO   0.500    0.267    0.500   0.383   0.267
Predicted AREA       24       48       48  36.000      24
Mask      WIDTH       8       15       15  11.500       8
          HEIGHT      4        4        4   4.000       4
Distance centroids:   Predicted:   40.226 Original: ---
```

(d) 21517.

Source: Author.

5.6 Model discussions

Table D.5 shows the result for the test set (and open eyes test set) for the trained models. As it can be seen, fine-tuning the UNet MobileNetV2 model has increased the Dice Score coefficient to a maximum of 0.8939 for the test set (and 0.8952 if only images with open eyes are considered).

The value is inferior to 0.90, but it should be noted that CelebAMask-HQ eyes cover significant less of the image compared to Zheng et al. (2022a) and Brummen et al. (2021).

Brummen et al. (2021) used a much smaller dataset, but contemplated higher resolution images focused on the periorbital region only. A limitation of the present study is that it did not consider dysmorphologies of the periorbital region, some of which were considered by Brummen et al. (2021). Their model is also more complex, as it uses a ResNet50 backbone. Table 5.10 provides a comparison between the ResNet50 and MobileNetV2 models.

Table 5.10: ResNet50 and MobileNetV2 Keras Applications deep learning pretrained models.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
ResNet50	98	74.9%	92.1%	25.6M	58.2	4.6
MobileNetV2	14	71.3%	90.1%	3.5M	25.9	3.8

Source: Keras (2022)

The top-1/5 accuracy refers to the model's performance on the ImageNet validation dataset.

Time per inference step is the average of 30 batches (batch size: 32) and 10 repetitions.

CPU: AMD EPYC Processor (with IBPB) (92 core), RAM: 1.7T, GPU: Tesla A100

The developed model can serve as a basis for aiding the generation of palpebral fissure datasets by providing pre-annotation. The trained UNet and LinkNet models are able to process images of any size, as long as the input shape (width and height) are divisible by 32.

The use of GPU makes the model application in low-cost/standard computers difficult. For computers used for gaming and graphics works, GPUs are more common and these are examples of a task where the subjects can be focused for extended periods of time. A GPU powered computer could also be used only to process videos taken from the subject.

6 CONCLUSION

The segmentation of palpebral fissures images through deep learning models on images where the eyes correspond to a small portion of the image was discussed. Images and segmentation masks generated based on the CelebAMask-HQ dataset were used for training, validation, and testing of deep learning based intelligent models. Characteristics of this dataset were discussed, and some erroneous base segmentations were identified (listed in the appendices E and F). Closed eyes images of the Closed Eyes in the Wild images were also used to investigate reducing the imbalance of the base dataset.

A review on physiological characteristics of the palpebral fissure was performed, which would allow, for example, the elaboration of techniques for automatic detection of failure in the segmentation process. The review on spontaneous blink characteristics guides the use of the proposed models for the detection of incomplete blinks.

Potential uses of these models have been discussed mainly in the context of blink completeness assessment, which could be a useful tool for CSV diagnosis (and with some developments, possibly a predictive action). Finally, this work also aims to raise awareness regarding computer vision syndrome.

The performance of the best model in terms of segmentation dice metric was similar to state-of-art works with a much smaller number of samples, but that contemplated higher resolution images focused on the periorbital region only. A limitation of the present study is that it did not consider dysmorphologies of the periorbital region.

The models presented, as developed, cannot be used directly in real time, preventing their use in the context of computer vision syndrome prevention. They can, however, be used as an aid in diagnosis from previously recorded videos of the user, this approach being especially interesting as this is a non-invasive approach and can be replicated under normal conditions of use.

The use of the appearance of the palpebral fissure obtained through the segmentation of palpebral fissures by intelligent models has the advantage of not requiring morphological operations of erosion and dilation to obtain the segmented image. Also, in relation to the use of the eye aspect ratio (EAR) metric (Soukupová, 2016), commonly used to monitor blinking and fatigue, the palpebral fissure aspect ratio has the advantage of having a value of 0 when the eyes are closed. This avoids the need for empirical or algorithmically determined determination of the threshold value, which varies by individual, to identify a complete blink. The determination of incomplete blinks, as in the case of EAR, still requires additional heuristics, such as temporal verification of the variation in the appearance of the palpebral fissure. Besides the fact that it is a robust metric, EAR is easy to compute and can be determined in real time. It is also invariant under in-plane rotations, which the presented version of the palpebral fissure aspect ratio is not. It is possible to add detection of the angle between the palpebral fissures to compensate for this effect, but this type of rotation is uncommon in the CelebAMask-HQ dataset as well as in computer use. Both metrics, EAR and palpebral fissure aspect ratio, lose their ability to discern eye status in the case of out-of-plane rotations. It was shown that for small horizontal rotations (about 10°), typical in front of a computer, this is not a problem. For moderate rotations, such as when using a second screen, estimating the position of the face allows the metric to be corrected, compensating for this effect.

Application of the segmentation model on video datasets such as Eyeblink8 can be performed to ascertain the model's ability to allow the detection of complete and incomplete blinks, and its performance can be compared with Fogelton, 2018. The reduction of the inference time of the models can also be investigated, as it would allow its use with CSV prevention character. Several approaches are possible, from reducing the size of the images (resizing or cropping to contain only the face or even the periorbital region), to reducing the complexity of the models. This can be done, for example, with a ResNet model with fewer layers, simply, or via pruning strategies.

The evolution of authentication techniques can benefit from the use of these models for the detection of the palpebral fissure region when the iris is not easily recognized. The extraction of the palpebral fissure and the determination of complete and incomplete blinks may also prove to be a useful mechanism for the detection of videos with manipulation of the face region (fake faces).

6.1 Future work

6.1.1 Dataset generation

Re-annotated the whole dataset by one annotator could be done to eliminate small inconsistencies (like an annotation bigger than the palpebral fissure) and fixing the masks with problems. This could lead to an improvement in the performance of the models, but the task would be time-consuming, especially for the first part (checking 30000 pairs of images of masks to ensure consistency in annotation).

Instead, problematic images and mask identified were discarded, as they represented only a small part of the dataset. Given the size of the dataset, the best models were able to not capture the small inconsistencies in data, given in general a good representation of the palpebral fissure when this was visible in the image.

The original images and segmentation masks were resized to 224 using *resize* of OpenCV with bilinear interpolation (INTER_LINEAR). Downscaling samples help in managing RAM and compute limitations, allowing mini-batch learnings and speeding up the training and inference times, but the resizing process may impact the performance of models as well. Talebi and Milanfar (2021) have shown that learned resizers can substantially improve performance of computer vision tasks.

Eyeblink8 annotations don't consider the whole face, as many times the forehead is missing. CelebAMask-HQ images, in contrast, contain many "complete" faces, with even hair and some surrounding features. The information of background can to be "distracting" to the model. A face detection algorithm (for example face detection using Haar Feature-based Cascade Classier) could be applied on CelebAMask-HQ original images, so the model can focus on the eyes in the face. This could have been an alternative to the pure resize approach when preprocessing the dataset. However, most likely, some faces would have to be manually cropped, specially with non-frontal faces.

Another strategy would be to rescale images to another sizes, like 256 x 256, and cropping random 224 x 224 patches, as done in Krizhevsky, Sutskever and Hinton (2012). This would have an artificial zooming effect, as bigger palpebral fissures would be available to the model.

Detection of faces with occlusion is one of the remaining challenges in the field of computer vision Alashbi (2021). Occlusion of part of the faces (excluding or including the eye region), that may be distracting for a landmark detector as pointed out by Soukupová and Cech (2016a), can be used as additional data augmentation to further improve the models. Part of face can be occluded by masks, as it was common during COVID-19 pandemic, garments, clothes, and accessories (hat, veil, hijab, niqab, ...). Hands are, by nature, one of the most frequent elements among occluders and, in some cases, they temporally cover one eye.

One limitation is that CelebAMask-HQ does not contain many dysmorphologies of the eye and periorbital region. This is also true for Closed Eyes in the Wild and thus this is may limit the use of the models for these cases.

6.1.2 Reducing inference time and complexity

Resizing the images to a smaller size is a way of reducing the training and inference time of the models. Computational complexity can be reduced by regionalizing the detection area, keeping only the face without hair and surroundings. Based in Hashemi (2019), zero-padding around smaller images, as opposed to interpolation, could be used to keep a fixed size before composing the batches that are used to training CNN.

Another approach would be to keep only the periocular region. This is an interesting option, as information like the distance between the eyes, which can be used to give a rough estimation of the user distance to the screen, is still available. However, in some context, the forehead and mouth may be helpful to judge the state of the eyes and if the frames are part of the blink.

Cropping the images of the eyes only can block an interesting application that is to use the distance between the centroids of the palpebral fissures to roughly estimate the distance of the user to the screen, which is an important factor in CVS.

Simpler architectures and backbones are also one possibility. For example, a ResNet10 backbone could be tested. Other improvements can be made using pruning, which is the gradual elimination of neurons and connections or layers (following some heuristic) until the attainment of a simpler yet adequately performing model, and quantization. The quantized models use lower-precision (e.g. 8-bit instead of 32-bit float), which brings improvements via model compression and latency reduction. Converting the models to the Open Neural Network Exchange (ONNX) format is also an option. According to Rath (2021), some versions of the OpenCV DNN module may present a better inference speed than TensorFlow, especially on Intel processors when running on the CPU instead of a GPU.

Liu et al. (2021) have proposed an eye state detection based on weight binarization CNN and Transfer Learning. The use of the binary network reduce the storage space and speeds up the computation. An average accuracy of 97.41% on CEW, comparable to state-of-art methods, was obtained by this approach that is faster than non-binary network counterparts.

6.1.3 Not real-time, the use of ensembles and Test Time Augmentation (TTA)

If the system is used for performing blink analysis studies or assisting in the diagnosis of CVS during normal use of the device, a real-time system is not strictly necessary. Videos of the user can be taken under actual usage conditions, and these can be later analyzed. The detection of the presence of incomplete blinking could be done at this point. This is especially useful for monitoring the individual for longer periods to avoid having the awareness of the measurement interfere with blink behavior.

With this hypothesis, the performance of the system can be improved by using for example an ensemble of distinct neural networks, intelligent models, and other algorithms and/or using Test Time Augmentation. Krizhevsky, Sutskever and Hinton (2012) used a Test Time Augmentation (TTA) approach when testing their model in the 2010 version of ILSVRC, applying translations to images of the test set to improve the results.

No matter how well the model performs and how well its hyperparameters are fine-tuned, there can be stagnation in its performance, and the combination of models and algorithms can (at a higher computational cost) deliver better results than an individual model, assuming uncorrelated errors. This is the idea of meta-classifiers (ensembles). A majority voting system can be assigned to decide the value of each pixel in the segmentation mask, or different weights can be assigned to each model, if all solutions are segmentations models.

6.1.4 Distillation

Further enhancements to the model and dataset (by adding more data augmentations transformations and images with both eyes closed or only one open) may enable its utilization in a distillation fashion (HINTON; VINYALS; DEAN, 2015): another simpler and with less latency model is trained to emulate the output of the more complex model (or ensemble of models, as in Bucila, Caruana and Niculescu-Mizil (2006)) instead of a pre-defined ground truth. This way, larger unlabeled datasets could be used, reducing the risk of overfitting as more diverse data is available. This is specially interesting because there are many datasets with a significant number of faces available (Liu et al. (2015), for example, has 202599 faces and 10177 identities, and others can be found in Gross (2005) and websites like <https://face-rec.org/databases/>), but this is not the case for segmentations of the eyes regions. Another model could be used to detect if there are eyes visible in the images, to further improve robustness.

Datasets like Bae et al. (2023) may be an interesting option to be investigated, as the pipeline of this synthetic dataset for face recognition have allowed controlling the distribution of the data to ensure a fair dataset, by addressing ethical issues (many datasets were collected without explicit consent) and data bias (celebrity faces images often are taken with strong lighting and make-up, also having imbalanced racial distribution). Gou et al. (2017) eye localization results were improved when a combination of real data and synthetic data was used.

6.1.5 Palpebral fissure aspect ratio determination alternatives

Manually annotated eye corners are available at Eyeblink8 and could be used instead of determining the width of the palpebral fissure. This would make the difference between the palpebral fissure aspect ratio of an open eye and a closed one more distinct.

When not disposing of manually annotations, automatically localized eye corners could be used, that may also serve another techniques in a metaclassifier approach. The detection of eye landmarks can be performed fast by face alignment. For images of Helen Facial Feature Dataset (LE et al., 2012), the landmark estimate can be done as fast as about one millisecond per image, according to Kazemi and Sullivan (2014).

6.1.6 On the use of batch normalizartion

Lozej et al. (2018), while working with semantic segmentation of the iris, found that the U-Net models of several depths not using batch normalization have performed slightly better than their counterparts. This could be an interesting point to be explored.

However, it should be noted that the study have used only 200 samples corresponding to 107 distinct subjects for training and testing (160 and 40 samples, respectively). The batch size was not specified, but it is possible that it did not represent well statistics of the actual dataset. Lange, Helfrich and Ye (2022) and Kolarik, Burget and Riha (2020) also indicate that batch normalization works best using large batch size during training. As the state-of-the-art segmentation convolutional neural network architectures may use considerable amounts of memory, large batch size is often impossible to achieve on current hardware.

In a blog post Leo (2022), while working with a deep learning eye tracking application, stated that updating batch normalization at inference time could further improve its results. This would be specially interesting as it may represent a form of calibration of the segmentation model to a new user as the frames are presented to the model.

Test-time batch-normalization and adaptation of batch-norm statistics were studied in Yang et al. (2022b) and Hu et al. (2021) and are a recent topic. Variations of the batch normalization layer were also studied recently (LANGE; HELFRICH; YE, 2022).

6.1.7 Hyperparameters optimization

Searching in a low-dimensional space is usually done with grid search, which is less practical in high-dimensional spaces. Because the importance of each parameter was not known a priori, random search was chosen to explore the hyperparameter space. Furthermore, random sampling helps find good candidates faster (or, as it was shown, shows that the candidates already chosen had a reasonable performance with pretraining) and has helped to show a performance trend between the different topologies and models.

More sophisticated choices for hyperparameter tuning are also available, with examples being Bayesian Optimization and Hyperband, a speed-up variation of random search with adaptive resource allocation and early-stopping (LI et al., 2018).

6.1.8 Pretraining

Yang et al. (2022a) have demonstrated that face obfuscation has minimal impact on the accuracy of recognition models for many tasks and, in special, that in many cases, features learned on obfuscated images are equally transferable. Using a dataset with face obfuscation for pretraining would still be useful, as the model would still achieve faster and better convergence by knowing simple shapes and even human-like palpebral fissures. This is, however, out of the scope of this research in the current stage.

6.1.9 Training

Other Cost functions (or combinations of loss functions) could be used during training. One example is focal loss, with a gamma value different of zero. An additional loss function penalizing the misclassification of closed images could be useful to further improve the models' ability to detect a fully closed eye.

6.1.10 Implementing an alert system based on palpebral fissure

Correctly considering non-blinks enhance system performance of blink detection for preventing or diagnosing CSV, as noted by Pal et al. (2014), that have also pointed out that it is less detrimental to detect CVS when it is absent than to no identify when it actually occurs. This point should guide decision of developers and system designers.

If the person is constantly looking down, the effects of the drying of the ocular surface may be reduced (TSUBOTA; NAKAMORI, 1993), partially reducing the effects of computer syndrome vision. Based on this, a system designer developing an application that alerts the user in case of a low count of blinks may consider that it is better to reckon this situation as an incomplete blink, if this gives a better performance in general, to make the system less intrusive.

Szczesna-Iskander and Quintana (2020) found that when a patient's blink is forced (unnaturally prolonged), noninvasive tear-film breakup time is statistically and clinically significantly shorter than that observed for close-to-natural blinking conditions. According to the authors, forced blinks seem to induce more abrupt tear-film destabilization than close-to-natural blinks, so precise instructions should be given to the subjects regarding the blink type because it substantially impacts the assessment of tear-film stability measurements and surface quality.

Szczesna-Iskander and Quintana (2020) research indicates that an application that alerts the user to blink may benefit of instructing the user to perform short blinks, instead of relying on a "compensation strategy" of long blinks that are not so effective.

Kim et al. (2021) indicates that there are effective trainings for increasing the efficacy of the blink. Once the presence of incomplete blinks is detected, a professional ophthalmologist may indicate these trainings.

REFERENCES

AHMAD, N. et al. An integrated approach for eye centre localization using deep networks and rectangular-intensity-gradient technique. **Journal of King Saud University - Computer and Information Sciences**, v. 34, n. 9, p. 7153–7167, out. 2022. ISSN 1319-1578. Available from: <https://www.sciencedirect.com/science/article/pii/S1319157822000489>.

ALASHBI, A. A. Deep-Learning-CNN for Detecting Covered Faces with Niqab. In: . [s.n.], 2021. Available from: <https://www.semanticscholar.org/paper/Deep-Learning-CNN-for-Detecting-Covered-Faces-with-Alashbi/6efefbd0f4defffa2f3b9eb0342a27790e8bf279>.

ALPARSLAN, K.; ALPARSLAN, Y.; BURLICK, M. Towards Evaluating Driver Fatigue with Robust Deep Learning Models. jul. 2020. Available from: <https://arxiv.org/abs/2007.08453v4>.

American Optometric Association. **The Effects of Computer Use on Eye Health and Vision**. 1997. Available from: <https://documents.aoa.org/Documents/optometrists/effects-of-computer-use.pdf>.

American Optometric Association. **Computer vision syndrome (Digital eye strain)**. n.d. Available from: <https://www.aoa.org/healthy-eyes/eye-and-vision-conditions/computer-vision-syndrome?sso=y>.

ANBESU, E. W.; LEMA, A. K. Prevalence of computer vision syndrome: a systematic review and meta-analysis. **Scientific Reports**, v. 13, n. 1, p. 1801, jan. 2023. ISSN 2045-2322. Number: 1 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/s41598-023-28750-6>.

ANKRUM, D. R. **Viewing Distance at Computer Workstations**. 1996. From WorkPlace Ergonomics, Sept./Oct. 1996, p. 10-12. Available from: <http://andrej.web.elte.hu/www.office-ergo.com/viewing-distance.htm>.

BAE, G. et al. Digiface-1m: 1 million digital face images for face recognition. In: IEEE. **2023 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [S.l.], 2023.

BARROS, A. C. F. et al. Astenopia em docentes universitários durante a pandemia da COVID-19. **Revista Brasileira de Oftalmologia**, v. 81, p. e0007, fev. 2022. ISSN 0034-7280, 1982-8551. Publisher: Sociedade Brasileira de Oftalmologia. Available from: <https://www.scielo.br/j/rbof/a/t7SGj7k3RhYjrQQvRPHx6Zf/>.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, n. 10, p. 281–305, 2012. Available from: <http://jmlr.org/papers/v13/bergstra12a.html>.

BOLELLI, F. et al. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. **IEEE Transactions on Image Processing**, v. 29, p. 1999–2012, 2020.

BRUMMEN, A. V. et al. PeriorbitAI: Artificial intelligence automation of eyelid and periorbital measurements. **American journal of ophthalmology**, v. 230, p. 285–296, out. 2021. ISSN 0002-9394. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8862636/>.

BUCILA, C.; CARUANA, R.; NICULESCU-MIZIL, A. Model compression. In: **Knowledge Discovery and Data Mining**. [s.n.], 2006. Available from: <https://www.cs.cornell.edu/~caruana/compression.kdd06.pdf>.

CHAURASIA, A.; CULURCIELLO, E. LinkNet: Exploiting encoder representations for efficient semantic segmentation. In: **2017 IEEE Visual Communications and Image Processing (VCIP)**. [S.l.: s.n.], 2017. p. 1–4.

Chinese Academy of Sciences Institute of Automation (CASIA). **Casia Iris V1.** 2003. National Laboratory of Pattern Recognition (NLPR). Available from: <http://biometrics.idealtest.org/dbDetailForUser.do?id=1#/>.

CHOLLET, F. Xception: Deep Learning with Depthwise Separable Convolutions. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017. p. 1800–1807. ISSN: 1063-6919.

CHOLLET, F. **Keras documentation: Transfer learning & fine-tuning**. 2020. Available from: <https://keras.io/guides/transfer_learning/>.

CHOLLET, F.; OMERNICK, M. **Working with preprocessing layers | TensorFlow Core**. 2021. Available from: <https://www.tensorflow.org/guide/keras/preprocessing_layers>.

CORTACERO, K.; FISCHER, T.; DEMIRIS, Y. RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environments. In: **2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)**. [s.n.], 2019. p. 1159–1168. ISSN: 2473-9944. Available from: <https://ieeexplore.ieee.org/document/9022030>.

CRNOVRSANIN, T.; WANG, Y.; MA, K.-L. Stimulating a blink: reduction of eye fatigue with visual stimulus. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2014. (CHI '14), p. 2055–2064. ISBN 978-1-4503-2473-1. Available from: <https://doi.org/10.1145/2556288.2557129>.

CRUZ, A. A. V. et al. Spontaneous Eyeblink Activity. **The Ocular Surface**, v. 9, n. 1, p. 29–41, jan. 2011. ISSN 1542-0124. Available from: <https://www.sciencedirect.com/science/article/pii/S1542012411700076>.

DAIN, S.; MCCARTHY, A.; CHAN-LING, T. Symptoms in VDU operators. **American journal of optometry and physiological optics**, v. 65, p. 162–7, abr. 1988.

DEMENTYEV, A.; HOLZ, C. DualBlink: A Wearable Device to Continuously Detect, Track, and Actuate Blinking For Alleviating Dry Eyes and Computer Vision Syndrome. **Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies**, v. 1, n. 1, p. 1:1–1:19, mar. 2017. Available from: <https://doi.org/10.1145/3053330>.

DENG, J. et al. ImageNet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255. ISSN: 1063-6919.

Deutsche Gesetzliche Unfallversicherung. **Bildschirm- und Büroarbeitsplätze - Leitfaden für die Gestaltung**. 2019. Available from: <https://publikationen.dguv.de/regelwerk/dguv-informationen/409/bildschirm-und-bueroarbeitsplaetze-leitfaden-fuer-die-gestaltung>.

DEWI, C. et al. Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. **PeerJ Computer Science**, v. 8, p. e943, abr. 2022. ISSN 2376-5992. Publisher: PeerJ Inc. Available from: <https://peerj.com/articles/cs-943>.

DOGANAY, F. et al. The association between ocular dominance and physiological palpebral fissure asymmetry. **Laterality**, Routledge, v. 22, n. 4, p. 412–418, 2017. PMID: 27461553. Available from: <https://doi.org/10.1080/1357650X.2016.1209212>.

DRUTAROVSKY, T.; FOGELTON, A. Eye Blink Detection Using Variance of Motion Vectors. In: AGAPITO, L.; BRONSTEIN, M. M.; ROTHER, C. (Ed.). **Computer Vision - ECCV 2014 Workshops**. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science), p. 436–448. ISBN 978-3-319-16199-0.

FANTE, R. G. Chapter 17 - reconstruction of the eyelids. In: BAKER, S. R. (Ed.). **Local Flaps in Facial Reconstruction (Second Edition)**. Second edition. Edinburgh: Mosby, 2007. p. 387–413. ISBN 978-0-323-03684-9. Available from: <https://www.sciencedirect.com/science/article/pii/B9780323036849500220>.

FOGELTON, A. **Eyeblink, blinking matters, software dry eye treatment**. 2018. Available from: <https://www.blinkingmatters.com/>.

FOGELTON, A.; BENESOVA, W. Eye blink detection based on motion vectors analysis. **Computer Vision and Image Understanding**, v. 148, p. 23–33, jul. 2016. ISSN 1077-3142. Available from: <https://www.sciencedirect.com/science/article/pii/S1077314216300054>.

FOGELTON, A.; BENESOVA, W. Eye blink completeness detection. **Computer Vision and Image Understanding**, v. 176-177, p. 78–85, nov. 2018. ISSN 1077-3142. Available from: <https://www.sciencedirect.com/science/article/pii/S107731421830287X>.

GAO, W. et al. The cas-peal large-scale chinese face database and baseline evaluations. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 38, n. 1, p. 149–161, 2007.

Google for Developers. **Regularization for Simplicity: Lambda | Machine Learning**. 2022. Available from: <https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/lambda>.

GOU, C. et al. A joint cascaded framework for simultaneous eye detection and eye state estimation. **Pattern Recognition**, v. 67, p. 23–31, jul. 2017. ISSN 0031-3203. Available from: <https://www.sciencedirect.com/science/article/pii/S0031320317300250>.

GOURIER, N. Estimating face orientation from robust detection of salient facial features. In: **Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK**. [S.l.: s.n.], 2004.

GRIPP, K. W. et al. **Handbook of Physical Measurements.** Oxford University Press, 2013. v. 3rd edition. ISBN 9780199935710. Available from: <https://search.ebscohost.com/login.aspx?direct=true&AuthType=shib&db=nlebk&AN=644757&lang=pt-br&scope=site&authtype=guest,shib&custid=s5837110&groupid=main&profile=eds>.

GROSS, R. Face databases. In: S.LI, A. (Ed.). **Handbook of Face Recognition**. New York: Springer, 2005.

HALL, B. D. et al. Elements of morphology: Standard terminology for the periorbital region. **American Journal of Medical Genetics Part A**, v. 149A, n. 1, p. 29–39, 2009. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ajmg.a.32597>.

HASHEMI, M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. **Journal of Big Data**, v. 6, n. 1, p. 98, nov. 2019. ISSN 2196-1115. Available from: <https://doi.org/10.1186/s40537-019-0263-7>.

HE, K. et al. Deep Residual Learning for Image Recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778. ISSN: 1063-6919.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. In: **NIPS Deep Learning and Representation Learning Workshop**. [s.n.], 2015. Available from: <http://arxiv.org/abs/1503.02531>.

HIROTA, M. et al. Effect of Incomplete Blinking on Tear Film Stability. **Optometry and Vision Science**, v. 90, n. 7, p. 650–657, jul. 2013. ISSN 1538-9235. Available from: <https://journals.lww.com/optvissci/Fulltext/2013/07000/Effect_of_Incomplete_Blinking_on_Tear_Film.6.aspx>.

HU, X. et al. **MixNorm: Test-Time Adaptation Through Online Normalization Estimation**. 2021.

HUANG, G. B. et al. **Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments**. [S.l.], 2007.

HUDA, C.; TOLLE, H.; UTAMININGRUM, F. Mobile-based driver sleepiness detection using facial landmarks and analysis of ear values. **International Journal of Interactive Mobile Technologies (iJIM)**, v. 14, n. 14, p. pp. 16–30, Aug. 2020. Available from: <https://online-journals.org/index.php/i-jim/article/view/14105>.

IAKUBOVSKII, P. **Tutorial — Fine Tuning**. 2018. Available from: <https://segmentation-models.readthedocs.io/en/1.0.1/tutorial.html#fine-tuning>.

IAKUBOVSKII, P. **Segmentation Models**. [S.l.]: GitHub, 2019. <https://github.com/qubvel/segmentation_models>.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: BACH, F.; BLEI, D. (Ed.). **Proceedings of**

the 32nd International Conference on Machine Learning. Lille, France: PMLR, 2015. (Proceedings of Machine Learning Research, v. 37), p. 448–456. Available from: <https://proceedings.mlr.press/v37/ioffe15.html>.

JASCHINSKI-KRUZA, W. Visual strain during vdu work: the effect of viewing distance and dark focus. **Ergonomics**, Taylor & Francis, v. 31, n. 10, p. 1449–1465, 1988. PMID: 3208736. Available from: <https://doi.org/10.1080/00140138808966788>.

JESORSKY, O.; KIRCHBERG, K. J.; FRISCHHOLZ, R. W. Robust Face Detection Using the Hausdorff Distance. In: BIGUN, J.; SMERALDI, F. (Ed.). **Audio- and Video-Based Biometric Person Authentication**. Berlin, Heidelberg: Springer, 2001. (Lecture Notes in Computer Science), p. 90–95. ISBN 978-3-540-45344-4.

KARRAS, T. et al. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. 2018.

KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2014. p. 1867–1874. ISSN: 1063-6919.

KENNARD, D. W.; SMYTH, G. L. Interaction of Mechanisms Causing Eye and Eyelid Movement. **Nature**, v. 197, n. 4862, p. 50–52, jan. 1963. ISSN 1476-4687. Number: 4862 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/197050a0>.

KERAS. **Keras documentation: Keras Applications**. 2022. Available from: <https://keras.io/api/applications/>.

KIM, A. D. et al. Therapeutic benefits of blinking exercises in dry eye disease. **Contact Lens and Anterior Eye**, v. 44, n. 3, p. 101329, jun. 2021. ISSN 1367-0484. Available from: <https://www.sciencedirect.com/science/article/pii/S1367048420300874>.

KIM, H. et al. Eye detection in a facial image under pose variation based on multi-scale iris shape feature. **Image and Vision Computing**, v. 57, p. 147–164, jan. 2017. ISSN 0262-8856. Available from: <https://www.sciencedirect.com/science/article/pii/S0262885616301858>.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2017.

KOLARIK, M.; BURGET, R.; RIHA, K. Comparing Normalization Methods for Limited Batch Size Segmentation Neural Networks. In: **2020 43rd International Conference on Telecommunications and Signal Processing (TSP)**. [S.l.: s.n.], 2020. p. 677–680.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Available from: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

KUWAHARA, A. et al. Blink Detection Using Image Processing to Predict Eye Fatigue. In: AHRAM, T. et al. (Ed.). **Human Interaction, Emerging Technologies and Future Applications III**. Cham: Springer International Publishing, 2021. (Advances in Intelligent Systems and Computing), p. 362–368. ISBN 978-3-030-55307-4.

KUWAHARA, A. et al. Eye Fatigue Prediction System Using Blink Detection Based on Eye Image. In: **2021 IEEE International Conference on Consumer Electronics (ICCE)**. [S.l.: s.n.], 2021. p. 1–3. ISSN: 2158-4001.

KUWAHARA, A. et al. Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping(EARM). **Cognitive Robotics**, v. 2, p. 50–59, jan. 2022. ISSN 2667-2413. Available from: <https://www.sciencedirect.com/science/article/pii/S2667241322000039>.

LANGE, S.; HELFRICH, K.; YE, Q. Batch normalization preconditioning for neural network training. **Journal of Machine Learning Research**, v. 23, n. 72, p. 1–41, 2022. Available from: <http://jmlr.org/papers/v23/20-1135.html>.

LE, V. et al. Interactive Facial Feature Localization. In: FITZGIBBON, A. et al. (Ed.). **Computer Vision – ECCV 2012**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 679–692. ISBN 978-3-642-33712-3.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, November 1998.

LECUN, Y. et al. Efficient backprop. In: ____. **Neural Networks: Tricks of the Trade**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 9–50. ISBN 978-3-540-49430-0. Available from: <https://doi.org/10.1007/3-540-49430-8_2>.

LEE, C.-H. et al. Maskgan: Towards diverse and interactive facial image manipulation. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2020.

LEE, H. et al. Development of Eye Blink Rate Level Classification System Utilizing Sitting Postural Behavior Data. **IEEE Access**, v. 9, p. 143677–143689, 2021. ISSN 2169-3536. Conference Name: IEEE Access.

LEE, W. O.; LEE, E. C.; PARK, K. R. Blink detection robust to various facial poses. **Journal of Neuroscience Methods**, v. 193, n. 2, p. 356–372, nov. 2010. ISSN 0165-0270. Available from: <https://www.sciencedirect.com/science/article/pii/S0165027010004942>.

LEO, O. **A Novel Way to Use Batch Normalization**. 2022. Available from: <https://towardsdatascience.com/a-novel-way-to-use-batch-normalization-837176d53525>.

LI, L. et al. Hyperband: A novel bandit-based approach to hyperparameter optimization. **Journal of Machine Learning Research**, v. 18, n. 185, p. 1–52, 2018. Available from: <http://jmlr.org/papers/v18/16-558.html>.

LI, Y.; CHANG, M.-C.; LYU, S. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In: **2018 IEEE International Workshop on Information Forensics and Security (WIFS)**. [S.l.: s.n.], 2018. p. 1–7. ISSN: 2157-4774.

LIU, Z. et al. Deep learning face attributes in the wild. In: **Proceedings of International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015.

LIU, Z.-T. et al. Eye state detection based on Weight Binarization Convolution Neural Network and Transfer Learning. **Applied Soft Computing**, v. 109, p. 107565, set. 2021. ISSN 1568-4946. Available from: <https://www.sciencedirect.com/science/article/pii/S1568494621004865>.

LOZEJ, J. et al. End-to-End Iris Segmentation Using U-Net. In: **2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)**. [S.l.: s.n.], 2018. p. 1–6.

LUCIO, D. R. et al. Fully Convolutional Networks and Generative Adversarial Networks Applied to Sclera Segmentation. In: **2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)**. [S.l.: s.n.], 2018. p. 1–7. ISSN: 2474-9699.

MAIOR, C. B. S. et al. Real-time classification for autonomous drowsiness detection using eye aspect ratio. **Expert Systems with Applications**, v. 158, p. 113505, nov. 2020. ISSN 0957-4174. Available from: <https://www.sciencedirect.com/science/article/pii/S0957417420303298>.

MCMONNIES, C. W. Diagnosis and remediation of blink inefficiency. **Contact Lens and Anterior Eye**, v. 44, n. 3, p. 101331, jun. 2021. ISSN 1367-0484. Available from: <https://www.sciencedirect.com/science/article/pii/S1367048420300989>.

MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: **2016 Fourth International Conference on 3D Vision (3DV)**. [S.l.: s.n.], 2016. p. 565–571.

MIN, C. et al. Tiger: Wearable Glasses for the 20-20-20 Rule to Alleviate Computer Vision Syndrome. In: **Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services**. New York, NY, USA: Association for Computing Machinery, 2019. (MobileHCI '19), p. 1–11. ISBN 978-1-4503-6825-4. Available from: <https://doi.org/10.1145/3338286.3340117>.

MONZO, D. et al. Precise eye localization using HOG descriptors. **Machine Vision and Applications**, v. 22, n. 3, p. 471–480, maio 2011. ISSN 1432-1769. Available from: <https://doi.org/10.1007/s00138-010-0273-0>.

MU, Y.; SUN, J.; HE, J. The Combined Focal Cross Entropy and Dice Loss Function for Segmentation of Protein Secondary Structures from Cryo-EM 3D Density maps. In: **2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.: s.n.], 2022. p. 3454–3461.

NOSCH, D. S. et al. Blink Animation Software to Improve Blinking and Dry Eye Symptoms. **Optometry and Vision Science**, v. 92, n. 9, p. e310, set. 2015. ISSN 1538-9235. Available from: <https://journals.lww.com/optvissci/Fulltext/2015/09000/Blink_Animation_Software_to_Improve_Blinking_and.26.aspx>.

NUTNICHA, N. et al. Provocation of dry eye disease symptoms during covid-19 lockdown. **Scientific Reports (Nature Publisher Group)**, v. 11, n. 1, 2021. Available from: <https://www.proquest.com/scholarly-journals/provocation-dry-eye-disease-symptoms-during-covid/docview/2613412043/se-2>.

Occupational Safety and Health Administration. **eTools : Computer Workstations - Workstation Components - Monitors | Occupational Safety and Health Administration**. n.d. Available from: <https://www.osha.gov/etools/computer-workstations/components/monitors>.

O'MALLEY, T. et al. **KerasTuner**. 2019. <https://github.com/keras-team/keras-tuner>.

PAL, M. et al. Electrooculography based blink detection to prevent computer vision syndrome. In: **2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)**. [S.l.: s.n.], 2014. p. 1–6.

PAN, G. et al. Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera. In: **2007 IEEE 11th International Conference on Computer Vision**. [S.l.: s.n.], 2007. p. 1–8. ISSN: 2380-7504.

PORTELLO, J. K.; ROSENFIELD, M.; CHU, C. A. Blink Rate, Incomplete Blinks and Computer Vision Syndrome. **Optometry and Vision Science**, v. 90, n. 5, p. 482–487, maio 2013. ISSN 1538-9235. Available from: <https://journals.lww.com/optvissci/Abstract/2013/05000/Blink_Rate,_Incomplete_Blinks_and_Computer_Vision.11.aspx>.

PRAKALAPAKORN, G. et al. **Dysmorphology of the Eye and Periorbital Region - EyeWiki**. 2023. Available from: <https://eyewiki.aao.org/Dysmorphology_of_the_Eye_and_Periorbital_Region>.

RANTI, C. et al. Blink Rate Patterns Provide a Reliable Measure of Individual Engagement with Scene Content. **Scientific Reports**, v. 10, n. 1, p. 8267, maio 2020. ISSN 2045-2322. Number: 1 Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/s41598-020-64999-x>.

RATH, S. **OpenCV DNN Module and Deep Learning (A Definitive guide)**. 2021. Available from: <https://learnopencv.com/deep-learning-with-opencvs-dnn-module-a-definitive-guide/>.

READ, S. A. et al. The Morphology of the Palpebral Fissure in Different Directions of Vertical Gaze. **Optometry and Vision Science**, v. 83, n. 10, p. 715, out. 2006. ISSN 1538-9235. Available from: <https://journals.lww.com/optvissci/Abstract/2006/10000/The_Morphology_of_the_Palpebral_Fissure_in.9.aspx>.

RODRIGUEZ, J. D. et al. Blink: Characteristics, Controls, and Relation to Dry Eyes. **Current Eye Research**, v. 43, n. 1, p. 52–66, jan. 2018. ISSN 0271-3683. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/02713683.2017.1381270. Available from: <https://doi.org/10.1080/02713683.2017.1381270>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: **Medical Image Computing and Computer-Assisted Intervention (MICCAI)**. Springer, 2015. (LNCS, v. 9351), p. 234–241. (available on arXiv:1505.04597 [cs.CV]). Available from: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>.

ROSENFIELD, M. Computer vision syndrome: a review of ocular causes and potential treatments. **Ophthalmic and Physiological Optics**, v. 31, n. 5, p. 502–515, 2011.

ISSN 1475-1313. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-1313.2011.00834.x. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-1313.2011.00834.x>.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, v. 115, n. 3, p. 211–252, dez. 2015. ISSN 1573-1405. Available from: <https://doi.org/10.1007/s11263-015-0816-y>.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 4510–4520.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. arXiv, 2014. Available from: <https://arxiv.org/abs/1409.1556>.

SINHA, S. **Image read and resize with OpenCV, Tensorflow and PIL.** 2020. Available from: <https://towardsdatascience.com/image-read-and-resize-with-opencv-tensorflow-and-pil-3e0f29b992be>.

SONG, F. et al. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. **Pattern Recognition**, v. 47, n. 9, p. 2825–2838, set. 2014. ISSN 0031-3203. Available from: <https://www.sciencedirect.com/science/article/pii/S0031320314001228>.

SOUKUPOVÁ, T.; CECH, J. Eye-blink detection using facial landmarks. In: . [S.l.: s.n.], 2016.

SOUKUPOVÁ, T.; CECH, J. Real-time eye blink detection using facial landmarks. In: . [S.l.: s.n.], 2016.

STERN, J. A.; WALRATH, L. C.; GOLDSTEIN, R. The endogenous eyeblink. **Psychophysiology**, v. 21, n. 1, p. 22–33, 1984. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.1984.tb02312.x>.

SU, Y. et al. Spontaneous Eye Blink Patterns in Dry Eye: Clinical Correlations. **Investigative Ophthalmology & Visual Science**, v. 59, n. 12, p. 5149–5156, out. 2018. ISSN 1552-5783. Available from: <https://doi.org/10.1167/iovs.18-24690>.

SUZUKI, S.; ABE, K. Topological structural analysis of digitized binary images by border following. **Computer Vision, Graphics, and Image Processing**, v. 30, n. 1, p. 32–46, 1985. ISSN 0734-189X. Available from: <https://www.sciencedirect.com/science/article/pii/0734189X85900167>.

SZCZESNA-ISKANDER, D. H.; QUINTANA, C. L. Subjective and Objective Evaluation of the Effect of Blink Type on Tear-film Breakup Time and Its Estimation. **Optometry and Vision Science**, v. 97, n. 11, p. 954–961, nov. 2020. ISSN 1538-9235. Available from: <https://journals.lww.com/optvissci/Fulltext/2020/11000/Subjective_and_Objective_Evaluation_of_the_Effect.6.aspx>.

TALEBI, H.; MILANFAR, P. Learning to Resize Images for Computer Vision Tasks. In: **2021 IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 487–496. ISSN: 2380-7504.

TODA, T.; NAKAI, M.; LIU, X. A close face-distance warning system for straightend neck prevention. In: **IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society**. [S.l.: s.n.], 2015. p. 003347–003352.

TORRALBA, A.; EFROS, A. A. Unbiased look at dataset bias. In: **CVPR 2011**. [S.l.: s.n.], 2011. p. 1521–1528. ISSN: 1063-6919.

TRUTOIU, L. C. et al. **Modeling and Animating Eye Blinks**. 2011. Available from: <https://la.disneyresearch.com/publication/modeling-and-animating-eye-blinks/>.

TSUBOTA, K.; NAKAMORI, K. Dry Eyes and Video Display Terminals. **New England Journal of Medicine**, v. 328, n. 8, p. 584–584, fev. 1993. ISSN 0028-4793. Publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJM199302253280817. Available from: <https://doi.org/10.1056/NEJM199302253280817>.

VASANTHAKUMAR, P.; KUMAR, P.; RAO, M. Anthropometric Analysis of Palpebral Fissure Dimensions and its Position in South Indian Ethnic Adults. **Oman Medical Journal**, v. 28, n. 1, p. 26–32, jan. 2013. ISSN 1999-768X. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3562989/>.

VIOLA, P.; JONES, M. J. Robust Real-Time Face Detection. **International Journal of Computer Vision**, v. 57, n. 2, p. 137–154, maio 2004. ISSN 1573-1405. Available from: <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>.

YAN, Z. et al. Computer Vision Syndrome: A widely spreading but largely unknown epidemic among computer users. **Computers in Human Behavior**, v. 24, n. 5, p. 2026–2042, set. 2008. ISSN 0747-5632. Available from: <https://www.sciencedirect.com/science/article/pii/S0747563207001501>.

YANG, K. et al. A study of face obfuscation in ImageNet. In: CHAUDHURI, K. et al. (Ed.). **Proceedings of the 39th International Conference on Machine Learning**. PMLR, 2022. (Proceedings of Machine Learning Research, v. 162), p. 25313–25330. Available from: <https://proceedings.mlr.press/v162/yang22q.html>.

YANG, T. et al. **Test-time Batch Normalization**. 2022.

YE, R. Z. et al. Effects of Image Quality on the Accuracy Human Pose Estimation and Detection of Eye Lid Opening/Closing Using Openpose and DLib. **Journal of Imaging**, v. 8, n. 12, p. 330, dez. 2022. ISSN 2313-433X. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute. Available from: <https://www.mdpi.com/2313-433X/8/12/330>.

YIN, Z. et al. Evaluation of VDT-Induced Visual Fatigue by Automatic Detection of Blink Features. **Sensors**, v. 22, n. 3, p. 916, jan. 2022. ISSN 1424-8220. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. Available from: <https://www.mdpi.com/1424-8220/22/3/916>.

YOKOO, S. **Carvana Image Masking Challenge**. 2017. Available from: <https://kaggle.com/competitions/carvana-image-masking-challenge>.

ZHAO, H. et al. Pyramid Scene Parsing Network. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017. p. 6230–6239. ISSN: 1063-6919.

ZHENG, Q. et al. Impact of Incomplete Blinking Analyzed Using a Deep Learning Model With the Keratograph 5M in Dry Eye Disease. **Translational Vision Science & Technology**, v. 11, n. 3, p. 38, mar. 2022. ISSN 2164-2591. Available from: <https://doi.org/10.1167/tvst.11.3.38>.

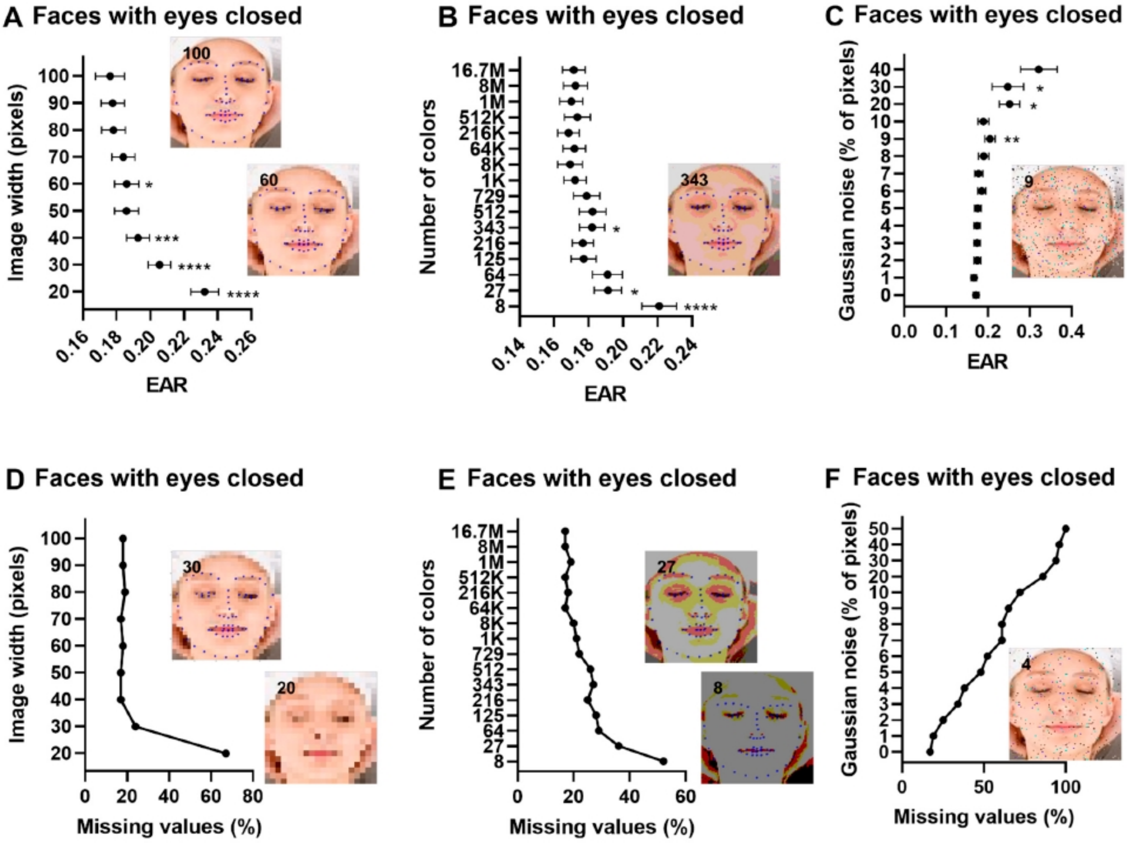
ZHENG, Q. et al. A texture-aware U-Net for identifying incomplete blinking from eye videography. **Biomedical signal processing and control**, v. 75, p. 103630, maio 2022. ISSN 1746-8094. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9484405/>.

APPENDIX A — EAR AND DLIB MODEL PERFORMANCE IN IMAGES AS FUNCTION OF IMAGE QUALITY

Ye et al. (2022) have analyzed the effects of image quality on the accuracy of human pose estimation and detection of eyelid opening/closing using Openpose and DLib. Concerning pretrained Dlib v19.24.0 models for facial landmark position, the rate of model failure remained acceptable at an image resolution of 60 x 60 pixels, a color depth of 343 colors, a light intensity of 14 lux, and a Gaussian noise level of 4% (i.e., 4% of pixels replaced by Gaussian noise).

The Figure A.1 and the Figure A.2 show EAR and model performance as a function of image quality using 100 images randomly selected each of the Closed Eyes in the Wild (CEW) dataset (SONG et al., 2014), with closed eyes and open eyes, respectively.

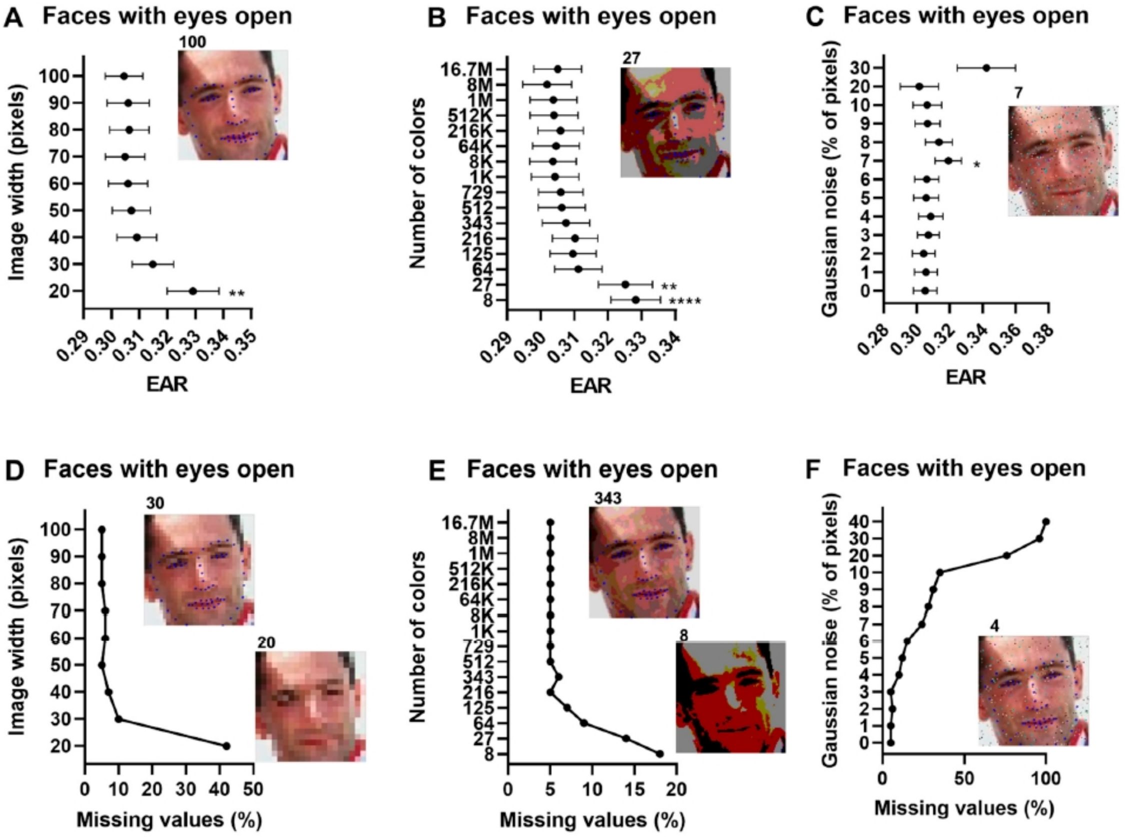
Figure A.1: EAR and model performance as a function of image quality using 100 images with closed eyes of the Closed Eyes in the Wild Dataset



EAR and model performance in the closed-eyes dataset as a function of image quality using the CEW Dataset (SONG et al., 2014). Panels (A) to (C) show the EAR estimates as a function of image resolution, color depth, and gaussian noise, respectively. Panels (D) to (F) show the percentage of missing values where the model failed to identify the face and/or both eyes as a function of image resolution, color depth, and gaussian noise, respectively. Inserts show images at different quality levels with overlaying model prediction. Data points ± error represent mean value ± standard error of the mean (SEM). Statistical significance levels were for one-way ANOVA with multiple comparisons using images of the best quality as the comparator. *: $P < 0.05$, **: $P < 0.01$; ***: $P < 0.001$; ***: $P < 0.0001$.

Source: Ye et al. (2022).

Figure A.2: EAR and model performance as a function of image quality using 100 images with open eyes of the Closed Eyes in the Wild Dataset

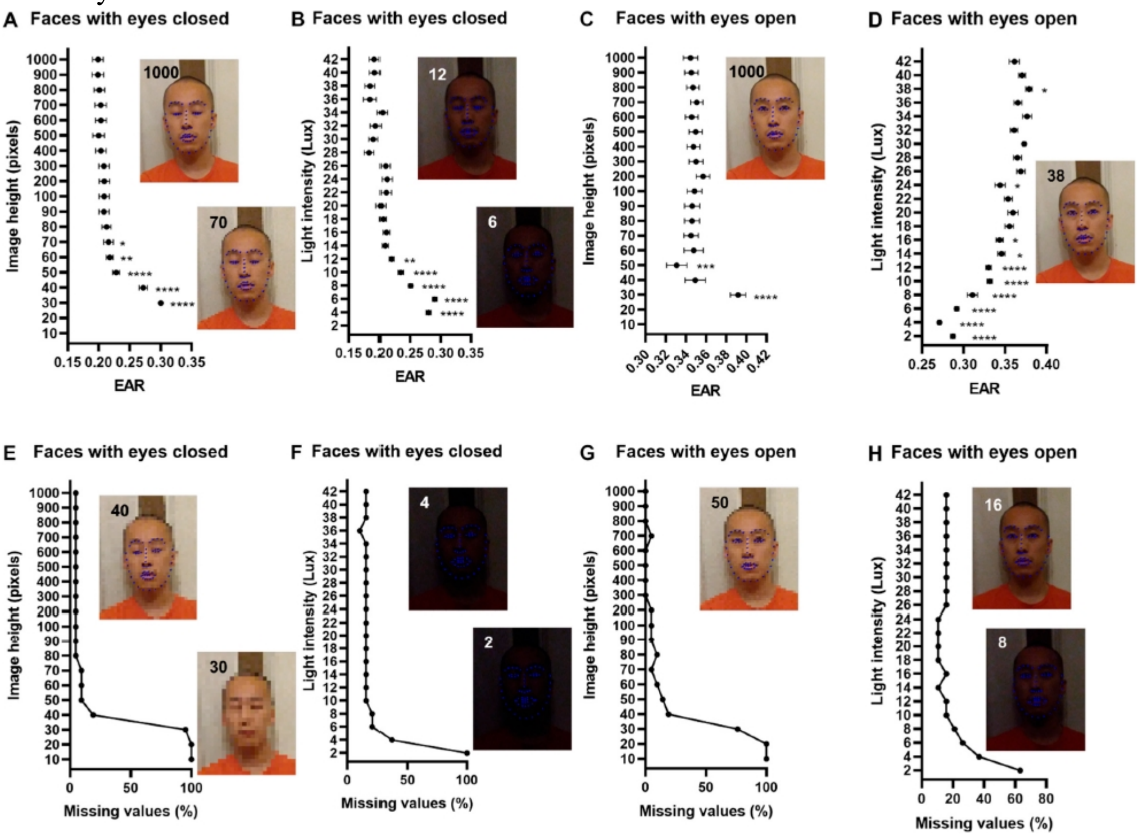


EAR and model performance in the open-eyes dataset as a function of image quality using the CEW Dataset (SONG et al., 2014). Panels (A) to (C) show the EAR estimates as a function of image resolution, color depth, and gaussian noise, respectively. Panels (D) to (F) show the percentage of missing values where the model failed to identify the face and/or both eyes as a function of image resolution, color depth, and gaussian noise, respectively. Inserts show images at different quality levels with overlaying model prediction. Data points ± error represent mean value ± standard error of the mean (SEM). Statistical significance levels were for one-way ANOVA with multiple comparisons using images of the best quality as the comparator. *: $P < 0.05$, **: $P < 0.01$; ***: $P < 0.001$; ***: $P < 0.0001$.

Source: Ye et al. (2022).

The Figure A.3 shows EAR and model performance as a function of image resolution and light intensity for 42 images captured using a smartphone camera (Moto E XT2052-1, 13 MP, f/2.0, 1/3.1), with the height of the face occupying approximately half of the image height. The light intensity at the level of the face was measured using a smartphone light meter application (Lux Meter (Light Meter), accessed on 15 July 2022).

Figure A.3: EAR and model performance as a function of image resolution and light intensity



EAR and model performance as a function of image resolution and light intensity. Panels (A) and (B) show the EAR estimates as a function of image resolution in faces with eyes closed. Panels (C) and (D) show the same in faces with eyes open. Panels (E) to (F) show the percentage of missing values where the model failed to identify the face and/or both eyes as a function of image resolution in faces with eyes closed. Panels (G) and (H) show the same in faces with eyes open. Inserts show images at different quality levels with overlaying model prediction. Data points ± error represent mean value ± standard error of the mean (SEM). Statistical significance levels were for one-way ANOVA with multiple comparisons using images of the best quality as the comparator. *: $P < 0.05$, **: $P < 0.01$; ***: $P < 0.001$; ***: $P < 0.0001$.

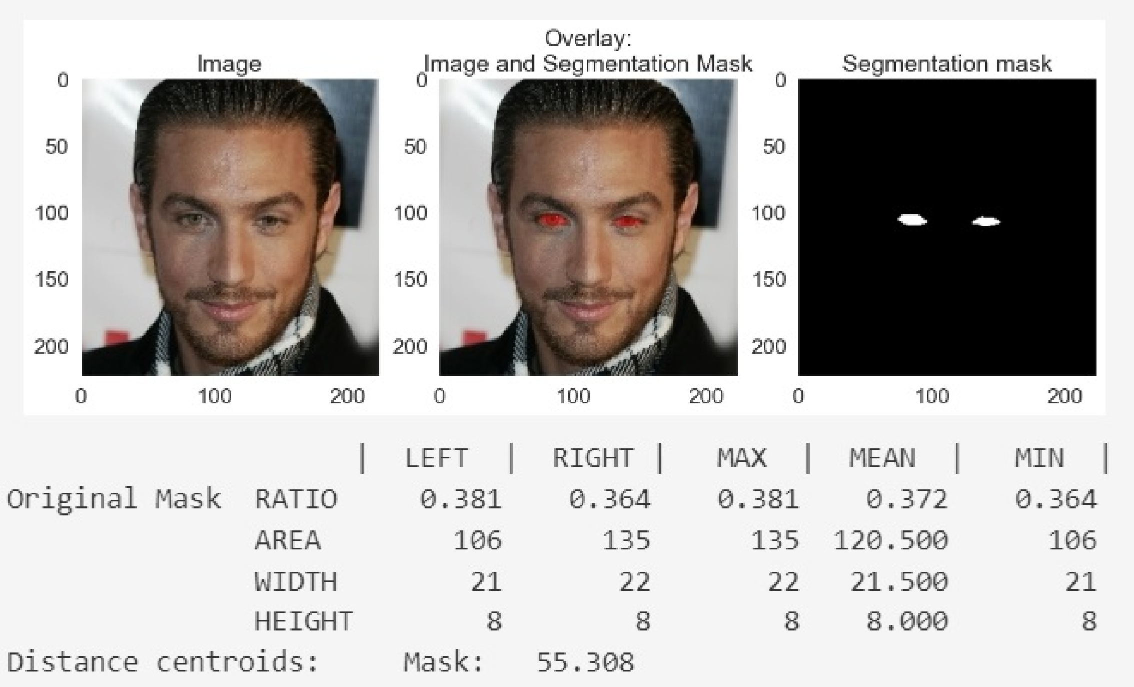
Source: Ye et al. (2022).

One interesting aspect is that the model positioning the landmarks seems to perform better with open eyes images, even with greater image quality, and even when only one person is considered.

APPENDIX B — EXAMPLES OF CONNECTED-COMPONENT ANALYSIS OF IMAGES WITH BOTH EYES OPEN FOR THE GENERATED DATASET

The Figure B.1 shows the output of the connected-component analysis (CCA) and contour detection for the sample 00030 of the generated dataset. The image, an overlay of the image and the segmentation mask, and the segmentation mask are shown. The image shows a typical case of the dataset, with a person with both eyes open, facing the camera (frontal head pose). There is palpebral fissure asymmetry, as discussed in subsection 2.1 Palpebral fissure, blink and computer use, but the area of the palpebral fissure and the ratio are relatively close.

Figure B.1: CCA of sample 00030

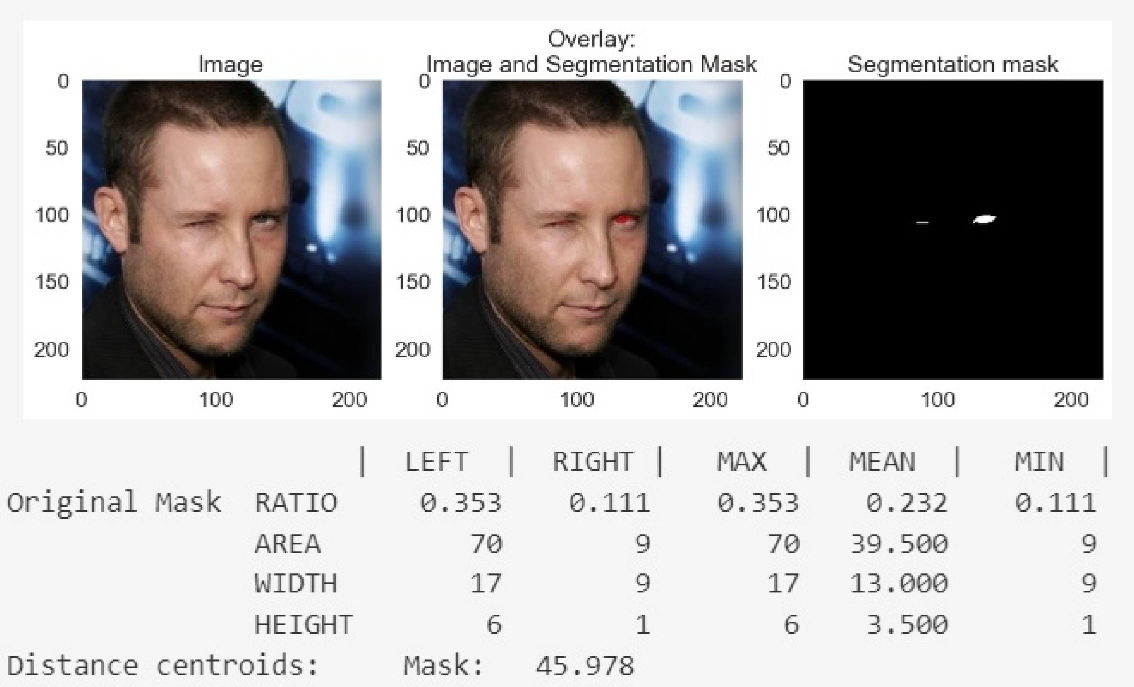


```
                        |  LEFT   |  RIGHT |    MAX   |  MEAN   |    MIN   |
Original Mask  RATIO       0.381     0.364    0.381     0.372     0.364
               AREA          106       135      135   120.500       106
               WIDTH          21        22       22    21.500        21
               HEIGHT          8         8        8     8.000         8
Distance centroids:     Mask:   55.308
```

Source: Author.

The Figure B.2 and Figure B.3 show the output of the CCA and contour detection for the samples 25986 and 14042, respectively, with moderate to strong out of the plane rotation. Figure B.2 shows a person squinting one eye and in a moderate out of the plane rotation (about approximately 20°). The yaw angle provokes a reduction of the distance of the centroids and of the width of palpebral fissure further from camera (increasing the ratio). Figure B.3 shows a person with a strong non-frontal head rotations (about approximately 60°). Here, we have a ratio of the height to width of 1. As the palpebral fissure aspect ratio was not defined in 3D, it loses discriminability in both cases. In the first situation, estimating the gaze direction may be a form to compensate for the rotation.

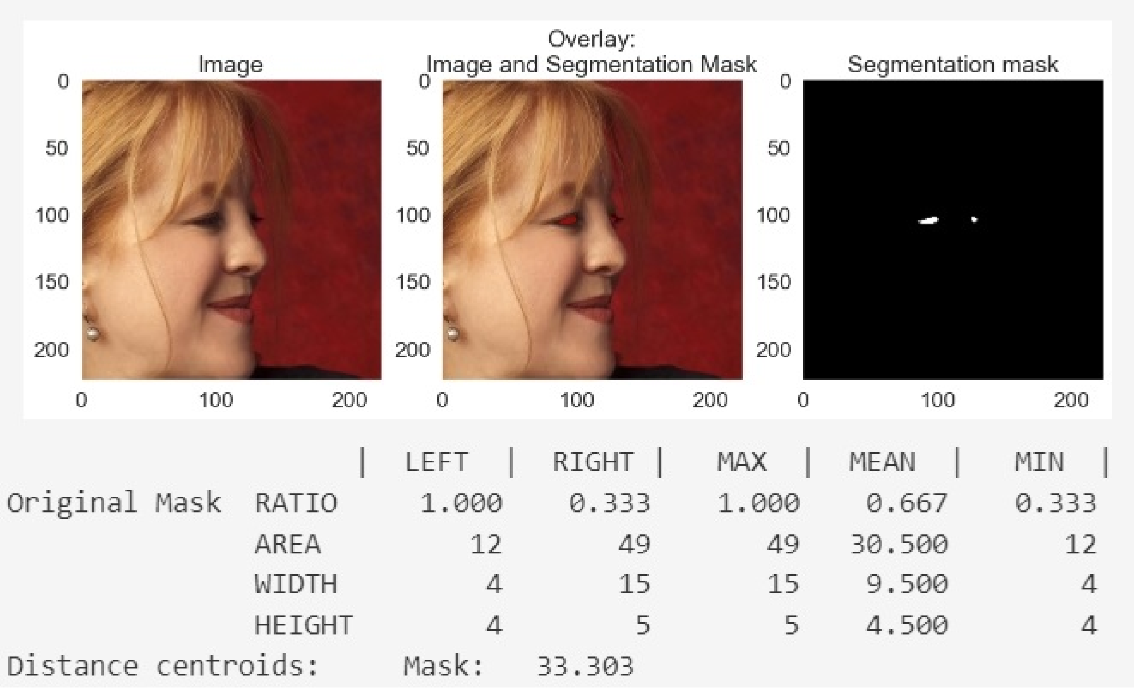
Figure B.2: CCA of sample 25986: person squinting one eye with head rotation



```
                    |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN  |
Original Mask  RATIO    0.353    0.111    0.353    0.232    0.111
               AREA        70        9       70   39.500        9
               WIDTH       17        9       17   13.000        9
               HEIGHT       6        1        6    3.500        1
Distance centroids:     Mask:   45.978
```

Source: Author.

Figure B.3: CCA of sample 14042: almost face profile

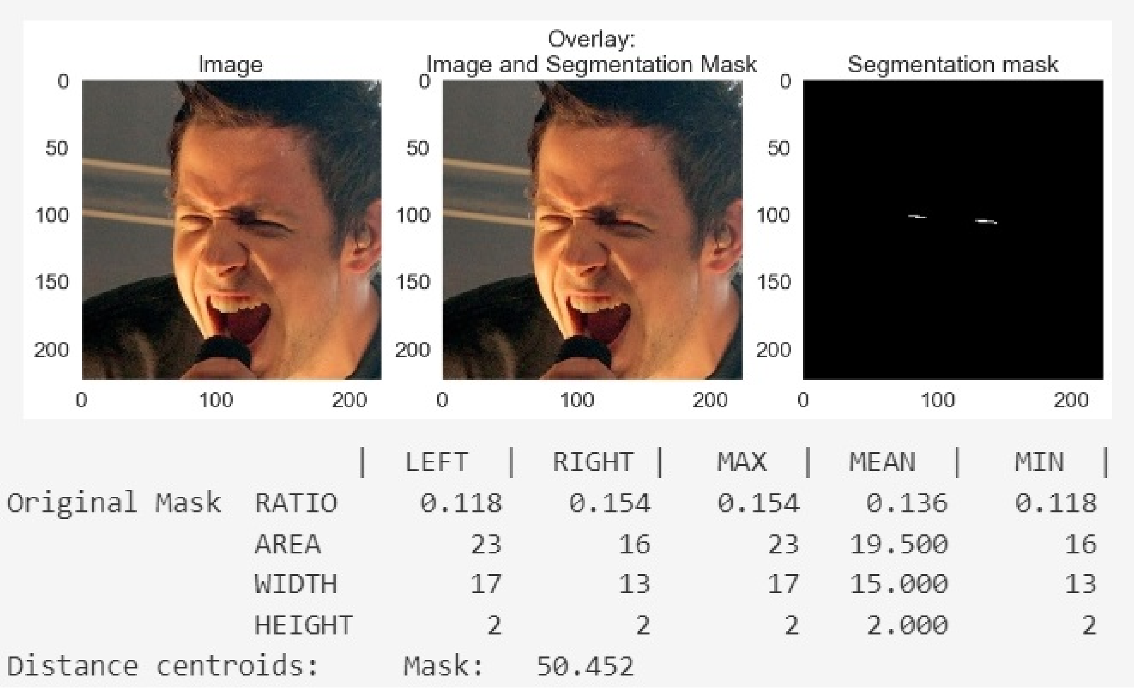


```
                    |  LEFT  |  RIGHT |   MAX  |  MEAN  |   MIN  |
Original Mask  RATIO    1.000    0.333    1.000    0.667    0.333
               AREA        12       49       49   30.500       12
               WIDTH        4       15       15    9.500        4
               HEIGHT       4        5        5    4.500        4
Distance centroids:     Mask:   33.303
```

Source: Author.

The Figure B.4, Figure B.6, and Figure B.5 show the output of the CCA and contour detection for the samples 06371 and 06874, respectively. These represent situations that may be particularly hard even for humans to define the state of the eye, or if the frame is part of a blink or not, without context. If the whole face is available, as in the proposed approach, the facial expression that is given to the model may help. Using temporal information about the opening and closing time for a blink event, as discussed in section 2.1.1 Eyeblink may also be useful in these situations.

Figure B.4: CCA of 06371: person squinting eyes with moderate head rotation



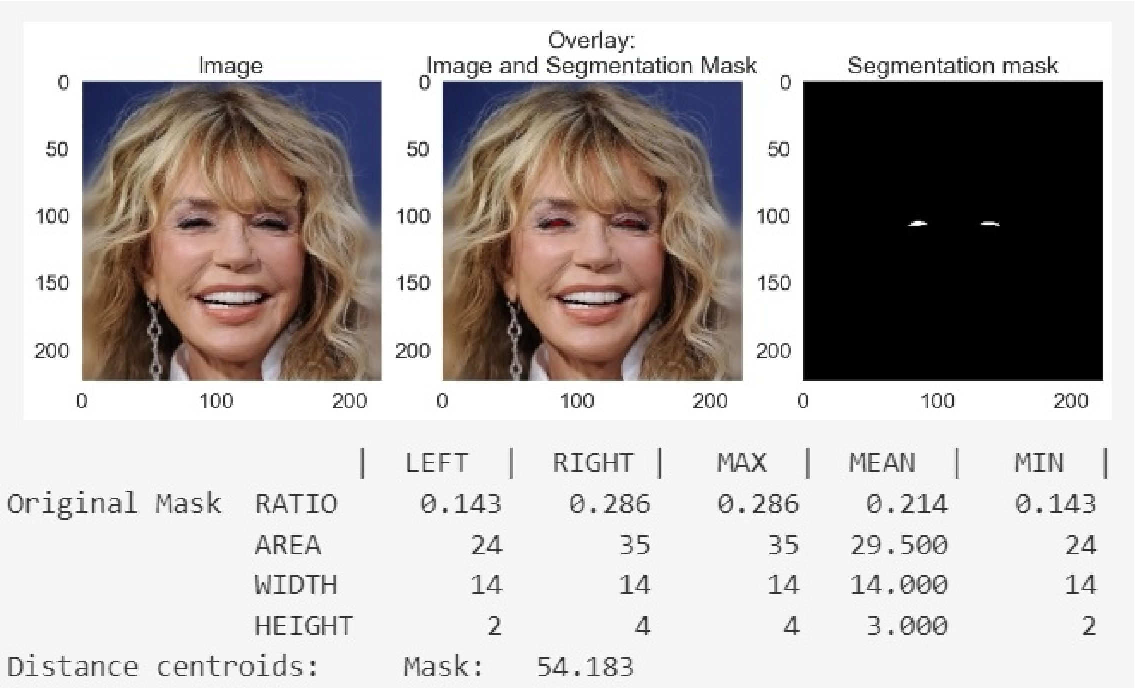
```
                     |  LEFT   |  RIGHT |    MAX   |  MEAN   |    MIN  |
Original Mask  RATIO      0.118    0.154     0.154    0.136     0.118
               AREA          23       16        23   19.500        16
               WIDTH         17       13        17   15.000        13
               HEIGHT         2        2         2    2.000         2
Distance centroids:     Mask:    50.452
```

Source: Author.

Figure B.4 also shows a moderate horizontal out of the plane rotation (about approximately 30°). The person appears to be singing, making a vivid facial expression with a wide open mouth.

Figure B.5 shows a person smiling, and with the image resized or in lower resolution, it is difficult to judge the state of the eyes, probably even more with only the periorbital region was available.

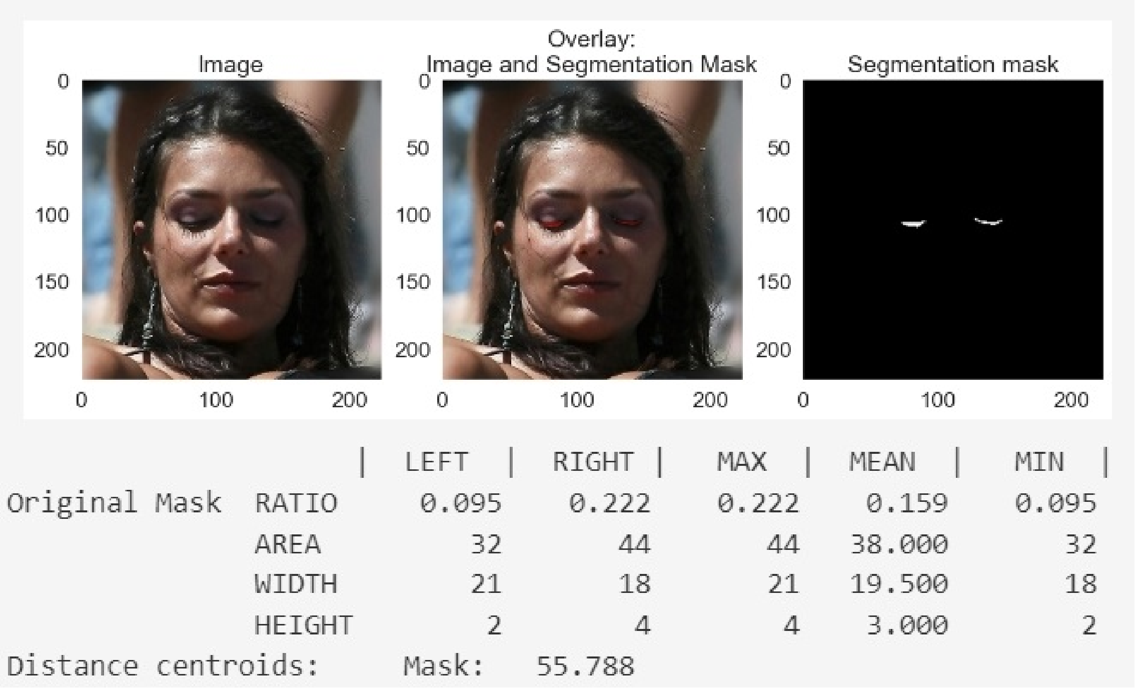
Figure B.5: CCA of sample 13708: person squinting eyes



```
                         |  LEFT   |  RIGHT |    MAX  |  MEAN   |   MIN  |
Original Mask  RATIO       0.143     0.286     0.286     0.214    0.143
               AREA           24        35        35    29.500       24
               WIDTH          14        14        14    14.000       14
               HEIGHT          2         4         4     3.000        2
Distance centroids:      Mask:   54.183
```

Source: Author.

Figure B.6: CCA of sample 06874: person looking downwards



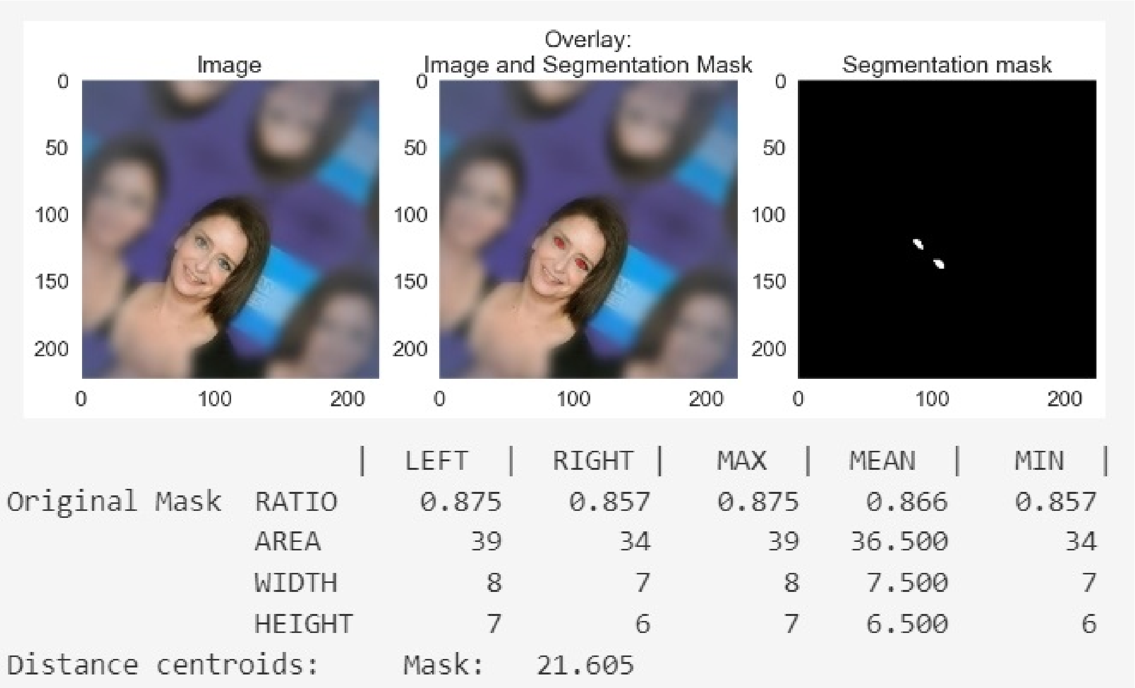
```
                         |  LEFT   |  RIGHT |    MAX  |  MEAN   |   MIN  |
Original Mask  RATIO       0.095     0.222     0.222     0.159    0.095
               AREA           32        44        44    38.000       32
               WIDTH          21        18        21    19.500       18
               HEIGHT          2         4         4     3.000        2
Distance centroids:      Mask:   55.788
```

Source: Author.

When the person has a downward gaze, as in Figure B.6, the palpebral fissure height, area, and ratio are rather small, when the person face is parallel to the camera. This situation can be confounded by a blink analysis system as an incomplete blink (or even a blink).

The Figure B.7 shows the output of the CCA and contour detection for the sample 08240 of the generated dataset sample. Pure in-plane rotation like in Figure B.7 are rare in the CelebAMask-HQ dataset. The most common form of rotation is out of the plane rotation.

Figure B.7: CCA of sample 08240: in-plane rotation (greater than 30°)



```
                     |  LEFT   |  RIGHT |   MAX   |  MEAN   |   MIN  |
Original Mask  RATIO     0.875    0.857    0.875    0.866    0.857
               AREA         39       34       39   36.500       34
               WIDTH         8        7        8    7.500        7
               HEIGHT        7        6        7    6.500        6
Distance centroids:     Mask:   21.605
```

Source: Author.

As in-plane rotation is not common in ordinary activity of a computer user, and 08240 eyes proportionally to the rest of the image are rather small compared to the rest of the dataset, the image was considered an outlier and was discarded.

APPENDIX C — WIDTH (IN PIXELS) OF THE PALPEBRAL FISSURE MULTIPLIED BY THE COSINE OF THE YAW ANGLE

Table C.1 shows the width (in pixels) of the palpebral fissure multiplied by the cosine of the yaw angle θ. Cells in the unfilled background region indicate unaffected values. The other regions indicate, starting in the region neighboring the unaffected one, 1 pixel difference (yellow), 2, 3, 4 and 5 or more pixels difference (shade of red, which occupies the region containing the lower right), respectively. For small angles, inferior to approximately 10°, the width is not affected.

Table C.1 could also reflect the effects of out-of-plane rotations of the head in the intraocular distance, if the value of the distance were rounded to an integer value. In this case, the yellow region goes one line up, to the row of 7.5°, for a width of 56 pixels. It should be noted, however, that for larger measures of length, the error of one or two pixels is less significant than it is for the palpebral fissure width. This is shown in Table C.2.

Even though Table C.1 and Table C.2 indicate yaw angles of 0° to 60°, observe that Occupational Safety and Health Administration (n.d.) indicates that monitors should not be farther than 35° degrees to the left or right. Also, it recommends that the center of the computer monitor should normally be located between 15% and 20% below the horizontal eye level and that the entire visual area of the display screen should be located in a manner that ensures that the downward viewing angle is never greater than 60°.

Table C.1: Width (in pixels) of the palpebral fissure multiplied by the cosine of the yaw angle θ

	Width (pixels)	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
θ(°)	cos θ																																												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
2.5	0.999	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
5	0.996	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
7.5	0.991	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
10	0.985	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	32	33	34	35	36	37	38	39	40	41	42	43	44
12.5	0.976	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
15	0.966	2	3	4	5	6	7	8	9	10	11	12	13	14	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	43
17.5	0.954	2	3	4	5	6	7	8	9	10	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	31	32	33	34	35	36	37	38	39	40	41	42	43
20	0.94	2	3	4	5	6	7	8	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	39	40	41	42
22.5	0.924	2	3	4	5	6	6	7	8	9	10	11	12	13	14	15	16	17	18	18	19	20	21	22	23	24	25	26	27	28	29	30	30	31	32	33	34	35	36	37	38	39	40	41	42
25	0.906	2	3	4	5	5	6	7	8	9	10	11	12	13	14	14	15	16	17	18	19	20	21	22	23	24	24	25	26	27	28	29	30	31	32	33	34	34	35	36	37	38	39	40	41
27.5	0.887	2	3	4	4	5	6	7	8	9	10	11	12	12	13	14	15	16	17	18	19	20	20	21	22	23	24	25	26	27	27	28	29	30	31	32	33	34	35	35	36	37	38	39	40
30	0.866	2	3	3	4	5	6	7	8	9	10	10	11	12	13	14	15	16	16	17	18	19	20	21	22	23	23	24	25	26	27	28	29	29	30	31	32	33	34	35	36	36	37	38	39
32.5	0.843	2	3	3	4	5	6	7	8	8	9	10	11	12	13	13	14	15	16	17	18	19	19	20	21	22	23	24	24	25	26	27	28	29	30	30	31	32	33	34	35	35	36	37	38
35	0.819	2	2	3	4	5	6	7	7	8	9	10	11	11	12	13	14	15	16	16	17	18	19	20	20	21	22	23	24	25	25	26	27	28	29	29	30	31	32	33	34	34	35	36	37
37.5	0.793	2	2	3	4	5	6	6	7	8	9	10	10	11	12	13	13	14	15	16	17	17	18	19	20	21	21	22	23	24	25	25	26	27	28	29	29	30	31	32	33	33	34	35	36
40	0.766	2	2	3	4	5	5	6	7	8	8	9	10	11	11	12	13	14	15	15	16	17	18	18	19	20	21	21	22	23	24	25	25	26	27	28	28	29	30	31	31	32	33	34	34
42.5	0.737	1	2	3	4	4	5	6	7	7	8	9	10	10	11	12	13	13	14	15	15	16	17	18	18	19	20	21	21	22	23	24	24	25	26	27	27	28	29	29	30	31	32	32	33
45	0.707	1	2	3	4	4	5	6	6	7	8	8	9	10	11	11	12	13	13	14	15	16	16	17	18	18	19	20	21	21	22	23	23	24	25	25	26	27	28	28	29	30	30	31	32
47.5	0.676	1	2	3	3	4	5	5	6	7	7	8	9	9	10	11	11	12	13	14	14	15	16	16	17	18	18	19	20	20	21	22	22	23	24	24	25	26	26	27	28	28	29	30	30
50	0.643	1	2	3	3	4	5	5	6	6	7	8	8	9	10	10	11	12	12	13	14	14	15	15	16	17	17	18	19	19	20	21	21	22	23	23	24	24	25	26	26	27	28	28	29
52.5	0.609	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10	10	11	12	12	13	13	14	15	15	16	16	17	18	18	19	19	20	21	21	22	23	23	24	24	25	26	26	27	27
55	0.574	1	2	2	3	3	4	5	5	6	6	7	7	8	9	9	10	10	11	11	12	13	13	14	14	15	15	16	17	17	18	18	19	20	20	21	21	22	22	23	24	24	25	25	26
57.5	0.537	1	2	2	3	3	4	4	5	5	6	6	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	16	16	17	17	18	18	19	19	20	20	21	21	22	23	23	24	24
60	0.5	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18	19	19	20	20	21	21	22	22	23

Cells in the unfilled background region indicate unaffected values. The other regions indicate, starting in the region neighboring the unaffected one, 1 pixel difference (yellow), 2, 3, 4 and 5 or more pixels difference, respectively. For small angles, inferior to approximately 10°, the width is not affected.

Source: Author.

Table C.2: Percentage error of width (in pixels) of the palpebral fissure multiplied by the cosine of the yaw angle θ compared to original witdh

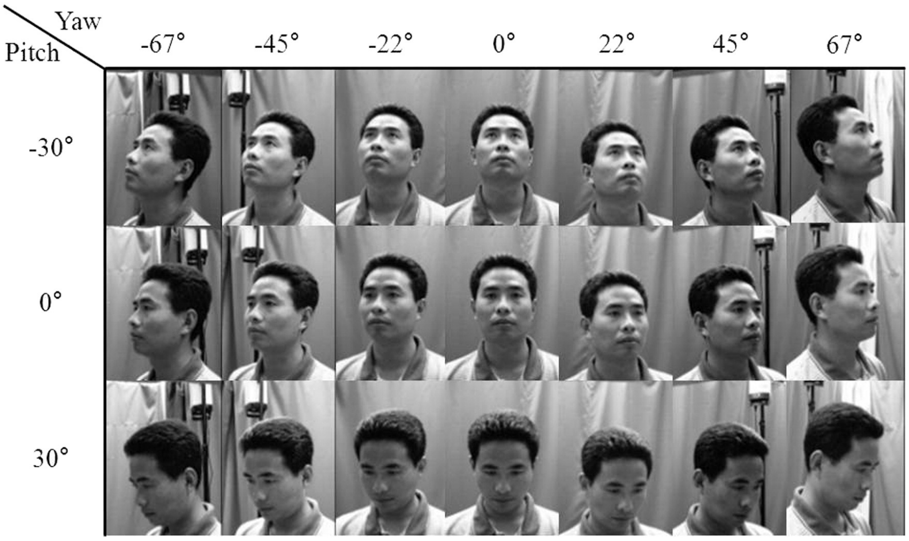
	Width (pixels)	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
$\theta(^\circ)$	$\cos\theta$																																												
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.5	0.999	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.996	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.5	0.991	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0.985	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	2	2	2	2	2
12.5	0.976	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2
15	0.966	0	0	0	0	0	0	0	0	0	0	0	0	0	7	6	6	6	5	5	5	5	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	4
17.5	0.954	0	0	0	0	0	0	0	0	0	9	8	8	7	7	6	6	6	5	5	5	5	4	4	4	4	4	4	3	3	3	3	6	6	6	6	5	5	5	5	5	5	5	5	4
20	0.94	0	0	0	0	0	0	0	11	10	9	8	8	7	7	6	6	6	5	5	5	5	4	4	4	8	7	7	7	7	6	6	6	6	6	6	5	5	5	5	5	7	7	7	7
22.5	0.924	0	0	0	0	0	14	13	11	10	9	8	8	7	7	6	6	6	5	10	10	9	9	8	8	8	7	7	7	7	6	6	9	9	9	8	8	8	8	8	7	7	7	7	7
25	0.906	0	0	0	0	17	14	13	11	10	9	8	8	7	7	13	12	11	11	10	10	9	9	8	8	8	11	11	10	10	10	9	9	9	9	8	8	11	10	10	10	10	9	9	9
27.5	0.887	0	0	0	20	17	14	13	11	10	9	8	8	14	13	13	12	11	11	10	10	9	13	13	12	12	11	11	10	10	13	13	12	12	11	11	11	11	10	13	12	12	12	11	11
30	0.866	0	0	25	20	17	14	13	11	10	9	17	15	14	13	13	12	11	16	15	14	14	13	13	12	12	15	14	14	13	13	13	12	15	14	14	14	13	13	13	12	14	14	14	13
32.5	0.843	0	0	25	20	17	14	13	11	20	18	17	15	14	13	19	18	17	16	15	14	14	17	17	16	15	15	14	17	17	16	16	15	15	14	17	16	16	15	15	15	17	16	16	16
35	0.819	0	33	25	20	17	14	13	22	20	18	17	15	21	20	19	18	17	16	20	19	18	17	17	20	19	19	18	17	17	19	19	18	18	17	19	19	18	18	18	17	19	19	18	18
37.5	0.793	0	33	25	20	17	14	25	22	20	18	17	23	21	20	19	24	22	21	20	19	23	22	21	20	19	22	21	21	20	19	22	21	21	20	19	22	21	21	20	20	21	21	20	20
40	0.766	0	33	25	20	17	29	25	22	20	27	25	23	21	27	25	24	22	21	25	24	23	22	25	24	23	22	25	24	23	23	22	24	24	23	22	24	24	23	23	24	24	23	23	24
42.5	0.737	50	33	25	20	33	29	25	22	30	27	25	23	29	27	25	24	28	26	25	29	27	26	25	28	27	26	25	28	27	26	25	27	26	26	25	27	26	26	28	27	26	26	27	27
45	0.707	50	33	25	20	33	29	25	33	30	27	33	31	29	27	31	29	28	32	30	29	27	30	29	28	31	30	29	28	30	29	28	30	29	29	31	30	29	28	30	29	29	30	30	29
47.5	0.676	50	33	25	40	33	29	38	33	30	36	33	31	36	33	31	35	33	32	30	33	32	30	33	32	31	33	32	31	33	32	31	33	32	31	33	32	32	33	33	32	33	33	32	33
50	0.643	50	33	25	40	33	29	38	33	40	36	33	38	36	33	38	35	33	37	35	33	36	35	38	36	35	37	36	34	37	35	34	36	35	34	36	35	37	36	35	37	36	35	36	36
52.5	0.609	50	33	50	40	33	43	38	44	40	36	42	38	36	40	38	41	39	37	40	38	41	39	38	40	38	41	39	38	40	39	41	39	38	40	39	38	39	38	40	39	38	40	39	40
55	0.574	50	33	50	40	50	43	38	44	40	45	42	46	43	40	44	41	44	42	45	43	41	43	42	44	42	44	43	41	43	42	44	42	41	43	42	43	42	44	43	41	43	42	43	42
57.5	0.537	50	33	50	40	50	43	50	44	50	45	50	46	43	47	44	47	44	47	45	48	45	48	46	48	46	48	46	45	47	45	47	45	47	46	47	46	47	46	48	46	45	47	45	47
60	0.5	50	33	50	40	50	43	50	44	50	45	50	46	50	47	50	47	50	47	50	48	50	48	50	48	50	48	50	48	50	48	50	48	50	49	50	49	50	49	50	49	50	49	50	49

Cells in the unfilled background region indicate unaffected values. 🟨 indicate percentage values inferior to 2.5%; 🟨, between 2.5% and 5%; 🟧, between 5% and 7.5%; 🟧, between 7.5% and 10%; 🟥, between 10% and 25%; 🟥, greater than 25%. For small angles, inferior to approximately 10°, the width is not affected.

Source: Author.

The Figure C.1 shows some examples of head pose variation in the CAS-PEAL database. The vertical head out of the plane rotations (pitch angle) variations are ±30° and 0° and the horizontal ones (yaw angle) are ±67°, ±45°, 22° and 0°.

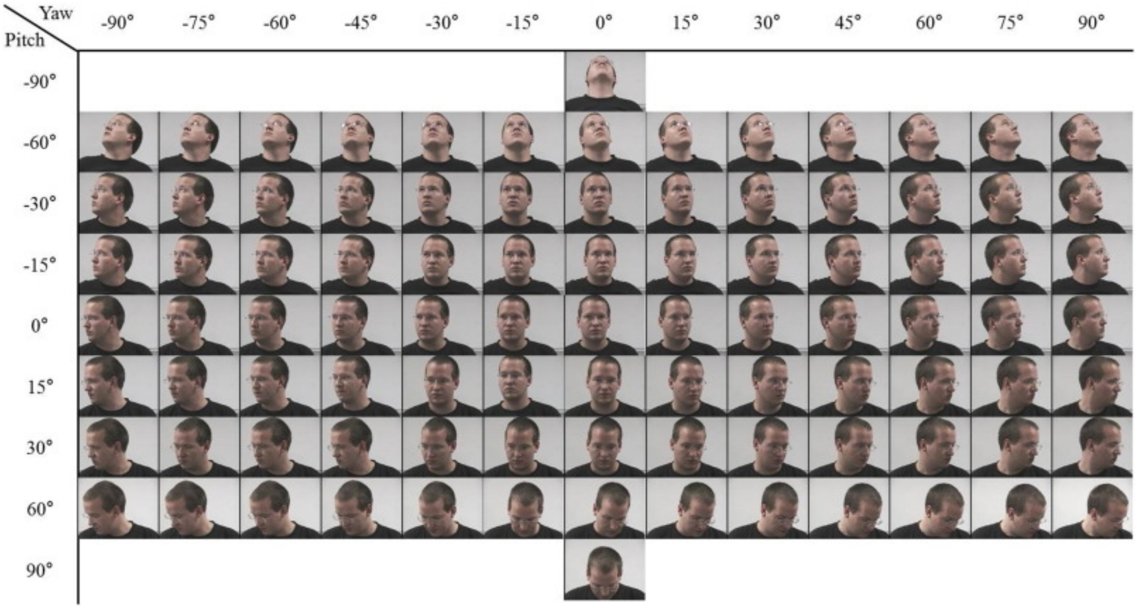
Figure C.1: Sample head pose images in the CAS-PEAL database



Source: image from Kim et al. (2017); CAS-PEAL database (GAO et al., 2007) samples.

The Figure C.2 shows some sample head pose images of the Pointing'04 database. In then, the pitch variations are from −60° to 60°, with two extreme poses, at ±90°. The yaw variations are from −90° to 90°.

Figure C.2: Sample head pose images of the Pointing'04 database



Source: image from Kim et al. (2017); Pointing'04 database (GOURIER, 2004) samples.

Figure C.1 and Figure C.2 (the reader is referred to the online high resolution version of the image available in Kim et al. (2017)) also suggest that, for a pitch angle close to 0°, the height of the palpebral fissure is unchanged to yaw angles up to close to 45°. The width of the eye further from camera reduces, making the ratio height/width to increase. Even if detecting complete blinks may not suffer loss in performance for a perfect model (as the predicted segmentation mask would be empty), detecting incomplete blinks would require some actions: only the greatest palpebral fissure to be considered, other features other than the palpebral fissure ratio to be used, the gaze angle to be estimated are some examples.

APPENDIX D — TABLES OF RESULTS OF EXPERIMENTS

All models were trained and evaluated in 2 CPUs Intel Xeon and GPU. The CPU usable memory is 12 GB. In almost all experiments covered here, the GPU Model is the Tesla T4. One experiment test is shown with GPU V100-SXM2-16GB, only to inform the inference time impact when changing the GPU.

The Table D.1 displays the results for the experiments with CelebAMask-HQ dataset, in descending order for the Dice score. The Table D.2 summarizes metrics values, while the Table D.3 displays the inference time experiments results.

The Table D.4 displays the results for the experiments with CelebAMask-HQ and CEW for pretraining.

The Table D.5 shows the result for the test set (and open eyes test set) for the trained models.

Table D.1: Results of metrics for pretraining with CelebAMask-HQ dataset only

Experiment Type	Model	Backbone	LR	L2 λ	Loss Value	Dice Score	IoU Score	Open Eyes Loss value	Open Eyes Dice Score	Open Eyes IoU Score	Best Epoch	Objective
Manual 1	UNet	MobileNetV2	0.0015	0.005	0.1214	0.8885	0.808	0.1152	0.8946	0.8134	5	bce_dice_loss
Manual 1	LinkNet	ResNet18	0.0015	0.005	0.1212	0.8878	0.8065	0.115	0.8927	0.8108	5	bce_dice_loss
Manual 3	UNet	MobileNetV2	0.0015	0.005	0.1204	0.8877	0.8073	0.1141	0.8935	0.8124	9	bce_dice_loss
Manual 3	LinkNet	ResNet18	0.0015	0.005	0.1234	0.8861	0.8042	0.1172	0.8911	0.8086	5	bce_dice_loss
Manual 2	UNet	MobileNetV2	0.0015	0.005	0.1247	0.8858	0.8045	0.1185	0.891	0.8091	5	bce_dice_loss
Manual 1	UNet	ResNet18	0.0015	0.005	0.1235	0.8852	0.8028	0.1173	0.8909	0.808	5	bce_dice_loss
Manual 2	UNet	ResNet18	0.0015	0.005	0.1249	0.8838	0.8008	0.1187	0.8898	0.8062	5	bce_dice_loss
Manual 2	LinkNet	ResNet18	0.0015	0.005	0.1263	0.8832	0.7993	0.1201	0.8888	0.8043	5	bce_dice_loss
Manual 3	UNet	ResNet18	0.0015	0.005	0.1262	0.8828	0.7996	0.1201	0.8883	0.8044	5	bce_dice_loss
Random 15 2	UNet	MobileNetV2	0.000211	0.016854	0.1315	0.8825	0.7993	0.1253	0.8874	0.8035	5	Dice (max)
Random 15 3	UNet	MobileNetV2	0.000211	0.016854	0.1371	0.8801	0.7953	0.131	0.8858	0.8003	5	Dice (max)
Random 15 1	UNet	MobileNetV2	0.000211	0.016854	0.1358	0.88	0.7949	0.1296	0.8857	0.7999	5	Dice (max)
Random 24	UNet	MobileNetV2	0.000136	0.090372	0.1349	0.8795	0.7941	0.1288	0.8832	0.7971	-	bce_dice_loss
Random 15 2	UNet	ResNet18	0.000333	0.015180	0.1337	0.8781	0.7924	0.1276	0.8837	0.7973	6	Dice (max)
Random 24	UNet	ResNet18	0.000923	0.022989	0.1343	0.8766	0.7911	0.1318	0.8781	0.7919	-	bce_dice_loss
Random 15 1	UNet	ResNet18	0.000333	0.015180	0.1382	0.8753	0.7878	0.1321	0.8813	0.7931	3	Dice (max)
Random 15 3	UNet	ResNet18	0.000333	0.015180	0.1396	0.8718	0.7838	0.1334	0.8773	0.7885	9	Dice (max)
Manual V100	LinkNet	MobileNetV2	0.0015	0.005	0.1534	0.8578	0.7625	0.1474	0.8631	0.7672	10	bce_dice_loss
Manual 3	LinkNet	MobileNetV2	0.0015	0.005	0.1554	0.8561	0.7603	0.1494	0.8615	0.765	5	bce_dice_loss
Manual 2	LinkNet	MobileNetV2	0.0015	0.005	0.1593	0.8527	0.7542	0.1533	0.8581	0.7589	10	bce_dice_loss
Random 24	LinkNet	ResNet18	0.000923	0.022989	0.1628	0.8511	0.7552	0.16	0.8525	0.7558	-	bce_dice_loss
Random 15 3	LinkNet	MobileNetV2	0.000923	0.022989	0.1614	0.8507	0.7529	0.1554	0.8553	0.7567	14	Dice (max)
Manual 1	LinkNet	MobileNetV2	0.0015	0.005	0.1626	0.8505	0.7523	0.1566	0.855	0.7561	10	bce_dice_loss
Random 15 2	LinkNet	MobileNetV2	0.000923	0.022989	0.1613	0.8504	0.7519	0.1554	0.8557	0.7565	10	Dice (max)

Table D.1 continued from previous page

Experiment Type	Model	Backbone	LR	L2 λ	Loss Value	Dice Score	IoU Score	Open Eyes Loss value	Open Eyes Dice Score	Open Eyes IoU Score	Best Epoch	Objective
Random 15 1	LinkNet	MobileNetV2	0.000923	0.022989	0.1618	0.8502	0.7507	0.1558	0.8555	0.7553	10	Dice (max)
Random 24	LinkNet	MobileNetV2	0.000420	0.062136	0.1788	0.8363	0.7357	0.1773	0.8378	0.7365	-	bce_dice_loss
Random 15 2	LinkNet	ResNet18	0.000353	0.034600	0.2039	0.8094	0.695	0.1983	0.814	0.6988	9	Dice (max)
Random 15 1	LinkNet	ResNet18	0.000353	0.034600	0.2156	0.7993	0.6806	0.2101	0.8033	0.6837	9	Dice (max)
Random 15 3	LinkNet	ResNet18	0.000353	0.034600	0.2212	0.7936	0.6729	0.2156	0.7984	0.6768	10	Dice (max)

Source: Author.

Table D.2: Metrics summary of experiments with only CelebAMask-HQ

Type	**Model**	**Backbone**	**Dice Score**			**IoU Score**		**Open eyes Dice Score**			**Open eyes IoU Score**	
			max	**mean**	**std**	**mean**	**std**	**max**	**mean**	**std**	**mean**	**std**
Manual	UNet	Mobile.V2	0.8885	0.8874	0.0014	0.8066	0.0019	0.8946	0.8930	0.0018	0.8116	0.0023
	LinkNet	ResNet18	0.8878	0.8857	0.0023	0.8033	0.0037	0.8927	0.8909	0.0020	0.8079	0.0034
	UNet	ResNet18	0.8852	0.8839	0.0012	0.8011	0.0016	0.8909	0.8897	0.0013	0.8062	0.0018
Random 15	UNet	Mobile.V2	0.8825	0.8809	0.0014	0.7965	0.0024	0.8874	0.8863	0.0010	0.8012	0.0020
Random	UNet	Mobile.V2	0.8795	0.8795	-	0.7941	-	0.8832	0.8832	-	0.7971	-
Random 15	UNet	ResNet18	0.8781	0.8751	0.0031	0.7880	0.0043	0.8837	0.8808	0.0033	0.7930	0.0044
Random	UNet	ResNet18	0.8766	0.8766	-	0.7911	-	0.8781	0.8781	-	0.7919	-
Manual	LinkNet	Mobile.V2	0.8561	0.8531	0.0028	0.7556	0.0041	0.8615	0.8582	0.0033	0.7600	0.0045
Random	LinkNet	ResNet18	0.8511	0.8511	-	0.7552	-	0.8525	0.8525	-	0.7558	-
Random 15	LinkNet	Mobile.V2	0.8507	0.8504	0.0003	0.7518	0.0011	0.8557	0.8555	0.0002	0.7562	0.0007
Random	LinkNet	Mobile.V2	0.8363	0.8363	-	0.7357	-	0.8378	0.8378	-	0.7365	-
Random 15	LinkNet	ResNet18	0.8094	0.8008	0.0079	0.6828	0.0112	0.8140	0.8052	0.0080	0.6864	0.0112

The experiment with GPU V100-SXM2-16GB was omitted here.

Source: Author.

Table D.3: Results of time inference for pretraining with CelebAMask-HQ

Experiment Type	Model	Backbone	LR	L2 λ	Dice Score	CPU Freq. (GHz)	Inference time 30 batches (size: 32, 10 runs)	Inference time for 1 image in batch	Inference time 960 images (10 runs): predict	Inference time 960 images (10 runs): call
Manual 1	UNet	Mobile.V2	0.0015	0.005	0.8885	2.3	3.54 s ± 910 ms	3.69 ± 0.948 ms	0.067 ± 0.021 s	0.110 ± 0.015 s
Manual 1	LinkNet	ResNet18	0.0015	0.005	0.8878	2	2.72 s ± 186 ms	2.83 ± 0.194 ms	0.073 ± 0.018 s	0.069 ± 0.009 s
Manual 3	UNet	Mobile.V2	0.0015	0.005	0.8877	2	3.82 s ± 646 ms	3.98 ± 0.673 ms	0.077 ± 0.052 s	0.118 ± 0.017 s
Manual 3	LinkNet	ResNet18	0.0015	0.005	0.8861	2	5.03 s ± 917 ms	5.24 ± 0.955 ms	0.072 ± 0.027 s	0.069 ± 0.010 s
Manual 2	UNet	Mobile.V2	0.0015	0.005	0.8858	2	4.98 s ± 902 ms	5.19 ± 0.94 ms	0.069 ± 0.027 s	0.115 ± 0.017 s
Manual 1	UNet	ResNet18	0.0015	0.005	0.8852	2	3.34 s ± 983 ms	3.48 ± 1.02 ms	0.066 ± 0.016 s	0.066 ± 0.009 s
Manual 2	UNet	ResNet18	0.0015	0.005	0.8838	2	3.72 s ± 1.06 s	3.88 ± 1.1 ms	0.068 ± 0.020 s	0.067 ± 0.010 s
Manual 2	LinkNet	ResNet18	0.0015	0.005	0.8832	2	4.61 s ± 1.02 s	4.80 ± 1.06 ms	0.068 ± 0.018 s	0.068 ± 0.018 s
Manual 3	UNet	ResNet18	0.0015	0.005	0.8828	2	2.76 s ± 185 ms	2.88 ± 0.193 ms	0.070 ± 0.018 s	0.065 ± 0.009 s
Random 15 2	UNet	Mobile.V2	0.000211	0.016854	0.8825	2	4.47 s ± 1.39 s	4.66 ± 1.45 ms	0.069 ± 0.022 s	0.109 ± 0.015 s
Random 15 3	UNet	Mobile.V2	0.000211	0.016854	0.8801	2.2	4.48 s ± 1.05 s	4.67 ± 1.09 ms	0.069 ± 0.022 s	0.110 ± 0.015 s
Random 15 1	UNet	Mobile.V2	0.000211	0.016854	0.88	2	4.27 s ± 973 ms	4.45 ± 1.01 ms	0.069 ± 0.025 s	0.113 ± 0.016 s
Random 24	UNet	Mobile.V2	0.000136	0.090372	0.8795	2.3	4.02 s ± 1.17 s	4.19 ± 1.22 ms	0.066 ± 0.026 s	0.108 ± 0.015 s
Random 15 2	UNet	ResNet18	0.000333	0.015180	0.8781	2	3.20 s ± 788 ms	3.33 ± 0.821 ms	0.070 ± 0.022 s	0.067 ± 0.010 s
Random 24	UNet	ResNet18	0.000923	0.022989	0.8766	2.3	3.83 s ± 1.2 s	3.99 ± 1.25 ms	0.069 ± 0.018 s	0.067 ± 0.010 s
Random 15 1	UNet	ResNet18	0.000333	0.015180	0.8753	2	4.62 s ± 1.04 s	4.81 ± 1.08 ms	0.069 ± 0.019 s	0.067 ± 0.010 s
Random 15 3	UNet	ResNet18	0.000333	0.015180	0.8718	2.2	3.96 s ± 1.14 s	4.13 ± 1.19 ms	0.067 ± 0.017 s	0.066 ± 0.009 s
Manual V100	LinkNet	Mobile.V2	0.0015	0.005	0.8578	2.2	1.43 s ± 365 ms	1.49 ± 0.38 ms	0.065 ± 0.022 s	0.124 ± 0.018 s
Manual 3	LinkNet	Mobile.V2	0.0015	0.005	0.8561	2	2.70 s ± 419 ms	2.81 ± 0.436 ms	0.062 ± 0.022 s	0.118 ± 0.016 s
Manual 2	LinkNet	Mobile.V2	0.0015	0.005	0.8527	2	3.44 s ± 1.1 s	3.58 ± 1.15 ms	0.065 ± 0.027 s	0.122 ± 0.018 s
Random 24	LinkNet	ResNet18	0.000923	0.022989	0.8511	2	2.68 s ± 340 ms	2.79 ± 0.354 ms	0.067 ± 0.018 s	0.076 ± 0.011 s
Random 15 3	LinkNet	Mobile.V2	0.000923	0.022989	0.8507	2.2	4.19 s ± 1.89 s	4.36 ± 1.97 ms	0.080 ± 0.078 s	0.124 ± 0.019 s
Manual 1	LinkNet	Mobile.V2	0.0015	0.005	0.8505	2	3.10 s ± 929 ms	3.23 ± 0.968 ms	0.071 ± 0.036 s	0.126 ± 0.017 s
Random 15 2	LinkNet	Mobile.V2	0.000923	0.022989	0.8504	2	3.49 s ± 1.49 s	3.64 ± 1.55 ms	0.070 ± 0.024 s	0.126 ± 0.021 s
Random 15 1	LinkNet	Mobile.V2	0.000923	0.022989	0.8502	2	2.86 s ± 341 ms	2.98 ± 0.355 ms	0.066 ± 0.023 s	0.119 ± 0.017 s

Table D.3 continued from previous page

Experiment Type	Model	Backbone	LR	L2 λ	Dice Score	CPU Freq. (GHz)	Inference time 30 batches (size: 32, 10 runs)	Inference time for 1 image in batch	Inference time 960 images (10 runs): predict	Inference time 960 images (10 runs): call
Random 24	LinkNet	Mobile.V2	0.000420	0.062136	0.8363	2.3	2.77 s ± 346 ms	2.89 ± 0.36 ms	0.082 ± 0.026 s	0.142 ± 0.035 s
Random 15 2	LinkNet	ResNet18	0.000353	0.034600	0.8094	2	2.75 s ± 441 ms	2.86 ± 0.459 ms	0.068 ± 0.018 s	0.075 ± 0.011 s
Random 15 1	LinkNet	ResNet18	0.000353	0.034600	0.7993	2	2.81 s ± 642 ms	2.93 ± 0.669 ms	0.069 ± 0.018 s	0.075 ± 0.010 s
Random 15 3	LinkNet	ResNet18	0.000353	0.034600	0.7936	2.2	2.79 s ± 727 ms	2.91 ± 0.757 ms	0.071 ± 0.024 s	0.077 ± 0.011 s

Source: Author.

Table D.4: Results for tests with CelebAMask-HQ and Closed Eyes in the Wild dataset for pretrained models

N	Model	Backbone	Loss Value	Dice Score	IoU Score	Op en Eyes Loss value	Open Eyes Dice Score	Open Eyes IoU Score	Best Epoch	Inference time 30 batches (size: 32, 10 runs)	Inference time 960 images (10 runs): predict	Inference time 960 images (10 runs): call
3	UNet	Mobil.V2	0.1205	0.8894	0.8091	0.1143	0.8940	0.8130	7	4.78 s ± 831 ms	0.070 ± 0.023 s	0.116 ± 0.016 s
2	LinkNet	ResNet18	0.1232	0.8855	0.8027	0.1216	0.8868	0.8034	14	3.25 s ± 895 ms	0.070 ± 0.021 s	0.068 ± 0.009 s
1	UNet	ResNet18	0.1242	0.8850	0.8023	0.1236	0.8851	0.8017	15	3.73 s ± 1.18 s	0.069 ± 0.018 s	0.070 ± 0.010 s
1	UNet	Mobil.V2	0.1255	0.8847	0.8033	0.1231	0.8859	0.8038	10	4.35 s ± 1.31 s	0.070 ± 0.023 s	0.115 ± 0.016 s
3	UNet	ResNet18	0.1263	0.8847	0.8016	0.1236	0.8851	0.8014	12	4.20 s ± 1.13 s	0.071 ± 0.018 s	0.072 ± 0.011 s
3	LinkNet	ResNet18	0.1250	0.8844	0.8018	0.1247	0.8842	0.8010	19	3.74 s ± 1.07 s	0.069 ± 0.019 s	0.069 ± 0.009 s
2	UNet	ResNet18	0.1256	0.8838	0.8012	0.1242	0.8845	0.8013	14	4.78 s ± 1.25 s	0.069 ± 0.018 s	0.069 ± 0.010 s
2	UNet	Mobil.V2	0.1269	0.8833	0.8007	0.1215	0.8868	0.8036	9	4.47 s ± 1.23 s	0.066 ± 0.022 s	0.111 ± 0.023 s
1	LinkNet	ResNet18	0.1270	0.8825	0.7988	0.1265	0.8824	0.7982	15	2.88 s ± 282 ms	0.071 ± 0.026 s	0.071 ± 0.010 s
2	LinkNet	Mobil.V2	0.1519	0.8590	0.7646	0.1459	0.8642	0.7691	9	2.80 s ± 296 ms	0.067 ± 0.029 s	0.119 ± 0.016 s
3	LinkNet	Mobil.V2	0.1547	0.8577	0.7627	0.1487	0.8619	0.7661	9	2.79 s ± 332 ms	0.065 ± 0.027 s	0.124 ± 0.017 s
1	LinkNet	Mobil.V2	0.1631	0.8489	0.7499	0.1571	0.8537	0.7540	7	2.77 s ± 292 ms	0.065 ± 0.023 s	0.124 ± 0.017 s

Source: Author.

Table D.5: Results of tests with CelebAMask-HQ and Closed Eyes in the Wild dataset for the trained models

N	Model	Backbone	Loss Value	Dice Score	IoU Score	Loss Value	Dice Score	IoU Score	Inference time 30 batches (size: 32, 10 runs)	Inference time for 1 image in batch	Inference time 960 images (10 runs): predict	Inference time 960 images (10 runs): call
2	UNet	MobileNetV2	0.1153	0.8939	0.8149	0.1125	0.8952	0.8157	4.28 ± 1.31 s	4.46 ± 1.36 ms	0.082 ± 0.025 s	0.132 ± 0.021 s
1	UNet	MobileNetV2	0.1165	0.8929	0.8139	0.1105	0.8971	0.8175	4.73 ± 1.19 s	4.93 ± 1.24 ms	0.084 ± 0.045 s	0.130 ± 0.022 s
3	UNet	MobileNetV2	0.1177	0.8918	0.8126	0.1123	0.8952	0.8154	4.07 ± 1.29 s	4.24 ± 1.34 ms	0.083 ± 0.024 s	0.132 ± 0.021 s

Source: Author.

APPENDIX E — CELEBAMASK-HQ DISCARDED SAMPLES

Table E.1 shows the discarded samples from the base dataset CelebAMask-HQ dataset (LEE et al., 2020):

Table E.1: CelebAMask-HQ dataset (LEE et al., 2020) discarded samples.

Sample	Observation
Problems with samples with no eyes and no glasses	
10219	One eye is visible
12633	Partial Visible eyes (noise)
19336	Visible eyes
23297	Visible eyes
Problems with samples with only left eye annotated	
07501	Eyeglass and only partial eye visible
15048	Both eyes visible; only left was annotated
20054	Both eyes visible; only left was annotated
25107	Both eyes visible; only left was annotated
25110	Both eyes visible; only left was annotated
25287	Both eyes visible; only left was annotated
28813	Left ear was annotated
Problems with samples with only right eye annotated	
01192	Both eyes visible; only right was annotated
06086	Both eyes visible; only right was annotated
16907	Both eyes visible; only right was annotated
17881	Both eyes visible; only right was annotated
28146	Wrong annotation
Problems with samples with both eyes annotated	
01840	Left ear was also annotated
02110	Outlier; eyes small
02807	Right eyebrow was also annotated
03177	Left ear was also annotated
06132	Left ear was also annotated
06136	Right ear was also annotated
06392	Right eyebrow was also annotated
07642	Right eyebrow was also annotated
07646	Left ear was also annotated
08181	Left eyebrow was also annotated
08240	Outlier; eyes small and in-plane head rotation
09956	Right ear was also annotated
14191	Right ear was also annotated
14400	Eyebrows were also annotated
14402	Eyebrows were also annotated
14531	Right eyebrow was also annotated
14698	Right eyebrow was also annotated
15537	Right ear was also annotated
16115	Right eyebrow was also annotated

Table E.1 continued from previous page

Sample	Observation
16122	Right eyebrow was also annotated
16264	Right eyebrow was also annotated
16637	Left ear was also annotated
16642	Left ear was also annotated
16823	Ears were also annotated
16993	Left eyebrow was also annotated
17239	Right ear was also annotated
18042	Left eyebrow was also annotated
18056	Right eyebrow was also annotated
18563	Right ear was also annotated
19008	Right eyebrow was also annotated
19120	Left ear was also annotated
19156	Left ear was also annotated
19262	Right eyebrow was also annotated
19381	Left ear was also annotated
20624	Left eyebrow was also annotated
20637	Left eyebrow was also annotated
21078	Left ear was also annotated
21492	Right ear was also annotated
21525	Right eyebrow was also annotated
23142	Outlier; eyes small
24035	Left ear was also annotated
24145	Right eyebrow was also annotated
26120	Left eyebrow was also annotated
27699	Right ear was also annotated
27923	Left ear was also annotated
28233	Left ear was also annotated

Source: Author.

APPENDIX F — CLOSED EYES IN THE WILD DISCARDED SAMPLES

Table F.1 shows the discarded samples from the base dataset Closed Eyes in the Wild dataset (SONG et al., 2014):

Table F.1: Closed Eyes in the Wild (SONG et al., 2014) discarded samples.

Sample	Original file name	Observation
30001	closed_eye_0002.jpg_face_2.jpg	One open eye from another person
30005	closed_eye_0012.jpg_face_1.jpg	Low resolution
30009	closed_eye_0019.jpg_face_1.jpg	Low resolution, min. image size 69
30010	closed_eye_0020.jpg_face_1.jpg	Partial open eyes
30011	closed_eye_0021.jpg_face_1.jpg	Low resolution
30015	closed_eye_0033.jpg_face_2.jpg	Partial open eyes
30016	closed_eye_0033.jpg_face_3.jpg	Low resolution, min. image size 66
30017	closed_eye_0034.jpg_face_4.jpg	Low resolution, min. image size 65
30019	closed_eye_0038.jpg_face_1.jpg	One partial open eye
30024	closed_eye_0059.jpg_face_2.jpg	Partial open eyes
30031	closed_eye_0074.jpg_face_1.jpg	Low resolution, person looking down
30036	closed_eye_0086.jpg_face_2.jpg	One partial open eye
30037	closed_eye_0087.jpg_face_1.jpg	One partial open eye
30038	closed_eye_0089.jpg_face_1.jpg	Partial open eyes
30046	closed_eye_0107.jpg_face_1.jpg	Low resolution, possible 1 open eye
30050	closed_eye_0132.jpg_face_1.jpg	Low resolution, min. image size 66
30051	closed_eye_0139.jpg_face_1.jpg	Low resolution
30065	closed_eye_0178.jpg_face_1.jpg	Partial open eyes
30069	closed_eye_0183.jpg_face_1.jpg	Low resolution, min. image size 60
30072	closed_eye_0189.jpg_face_1.jpg	Partial open eyes
30073	closed_eye_0189.jpg_face_2.jpg	Partial open eyes
30079	closed_eye_0207.jpg_face_1.jpg	Partial open eyes
30080	closed_eye_0207.jpg_face_2.jpg	Low resolution
30085	closed_eye_0218.jpg_face_1.jpg	One partial open eye
30090	closed_eye_0232.jpg_face_2.jpg	One partial open eye
30092	closed_eye_0237.jpg_face_1.jpg	Low resolution
30095	closed_eye_0243.jpg_face_1.jpg	One partial open eye
30098	closed_eye_0247.jpg_face_2.jpg	One partial open eye
30100	closed_eye_0249.JPG_face_1.jpg	Low resolution
30102	closed_eye_0251.jpg_face_1.jpg	One partial open eye
30103	closed_eye_0253.jpg_face_3.jpg	Low resolution, min. image size 69
30111	closed_eye_0276.jpg_face_2.jpg	Low resolution
30112	closed_eye_0279.jpg_face_2.jpg	Low resolution, min. image size 71
30113	closed_eye_0280.jpg_face_2.jpg	Low resolution
30118	closed_eye_0296.jpg_face_3.jpg	Low resolution
30120	closed_eye_0302.jpg_face_2.jpg	One partial open eye
30125	closed_eye_0315.jpg_face_1.jpg	One partial open eye
30126	closed_eye_0318.jpg_face_1.jpg	Low resolution
30132	closed_eye_0336.jpg_face_1.jpg	Low resolution, person looking down

Table F.1 continued from previous page

Sample	Original file name	Observation
30133	closed_eye_0341.jpg_face_2.jpg	Partial open eyes
30135	closed_eye_0344.jpg_face_2.jpg	Low resolution, possible 1 open eye
30137	closed_eye_0346.jpg_face_1.jpg	Low resolution
30138	closed_eye_0347.jpg_face_2.jpg	One open eye from another person
30139	closed_eye_0348.jpg_face_5.jpg	One open eye from another person
30147	closed_eye_0374.jpg_face_1.jpg	Low resolution
30150	closed_eye_0380.jpg_face_1.jpg	Low resolution
30154	closed_eye_0397.jpg_face_1.jpg	Low resolution
30155	closed_eye_0397.jpg_face_2.jpg	Low resolution
30158	closed_eye_0402.jpg_face_1.jpg	Low resolution
30160	closed_eye_0409.jpg_face_1.jpg	Low resolution, possible 1 open eye
30162	closed_eye_0418.jpg_face_2.jpg	Low resolution, possible 1 open eye
30175	closed_eye_0460.jpg_face_1.jpg	Possible looking down
30179	closed_eye_0469.jpg_face_1.jpg	Low resolution
30181	closed_eye_0476.jpg_face_1.jpg	Possible looking down
30185	closed_eye_0489.jpg_face_2.jpg	Low resolution, image effect
30186	closed_eye_0492.jpg_face_1.jpg	Drawing if 1 closed eye not visible
30187	closed_eye_0493.jpg_face_1.jpg	Partial open eyes
30188	closed_eye_0494.jpg_face_1.jpg	Low resolution
30202	closed_eye_0554.jpg_face_3.jpg	Low resolution
30211	closed_eye_0573.jpg_face_2.jpg	<- UNSURE, looking down?
30224	closed_eye_0619.jpg_face_1.jpg	Low resolution, possible 1 open eye
30226	closed_eye_0621.jpg_face_2.jpg	Low resolution, min. image size 64
30227	closed_eye_0622.jpg_face_1.jpg	Low resolution, min. image size 63
30228	closed_eye_0625.jpg_face_2.jpg	Low resolution
30229	closed_eye_0627.jpg_face_1.jpg	Low resolution
30231	closed_eye_0636.jpg_face_1.jpg	Open eyes
30232	closed_eye_0638.jpg_face_1.jpg	One open eye from another person
30236	closed_eye_0644.jpg_face_1.jpg	Partial open eyes
30241	closed_eye_0656.jpg_face_1.jpg	Low resolution, min. image size 62
30244	closed_eye_0665.jpg_face_1.jpg	Low resolution
30245	closed_eye_0666.jpg_face_1.jpg	Low resolution
30252	closed_eye_0682.jpg_face_3.jpg	One partial open eye
30258	closed_eye_0693.jpg_face_2.jpg	Low resolution, min. image size 71
30259	closed_eye_0693.jpg_face_3.jpg	One open eye
30261	closed_eye_0696.jpg_face_4.jpg	Low resolution
30262	closed_eye_0698.jpg_face_1.jpg	Partial open eyes
30273	closed_eye_0735.jpg_face_4.jpg	One partial open eye
30274	closed_eye_0736.jpg_face_1.jpg	Low resolution, possible 1 open eye
30276	closed_eye_0740.jpg_face_1.jpg	Low resolution
30283	closed_eye_0754.jpg_face_1.jpg	Low resolution
30284	closed_eye_0766.jpg_face_1.jpg	Low resolution
30285	closed_eye_0767.jpg_face_2.jpg	Low resolution, possible 1 open eye
30286	closed_eye_0767.jpg_face_3.jpg	One partial open eye
30287	closed_eye_0769.jpg_face_3.jpg	Low resolution, possible 1 open eye

Table F.1 continued from previous page

Sample	Original file name	Observation
30290	closed_eye_0772.jpg_face_2.jpg	One partial open eye
30292	closed_eye_0778.jpg_face_1.jpg	Low resolution, min. image size 55
30293	closed_eye_0780.jpg_face_1.jpg	One partial open eye
30298	closed_eye_0788.jpg_face_2.jpg	One open eye from another person
30311	closed_eye_0823.jpg_face_2.jpg	One open eye from another person
30314	closed_eye_0833.jpg_face_1.jpg	One partial open eye
30315	closed_eye_0836.jpg_face_1.jpg	Low resolution
30317	closed_eye_0841.jpg_face_2.jpg	Low resolution
30318	closed_eye_0843.jpg_face_1.jpg	Low resolution, min. image size 69
30319	closed_eye_0847.jpg_face_1.jpg	Low resolution
30326	closed_eye_0860.jpg_face_1.jpg	Low resolution
30329	closed_eye_0864.jpg_face_1.jpg	Low resolution, min. image size 63
30330	closed_eye_0865.jpg_face_1.jpg	Low resolution
30335	closed_eye_0880.jpg_face_1.jpg	Low resolution
30345	closed_eye_0894.jpg_face_1.jpg	Low resolution, possible 1 open eye
30348	closed_eye_0900.jpg_face_2.jpg	One open eye from another person
30352	closed_eye_0906.jpg_face_1.jpg	UNSURE, looking down?
30355	closed_eye_0917.jpg_face_3.jpg	One partial open eye
30356	closed_eye_0918.jpg_face_1.jpg	Low resolution
30357	closed_eye_0918.jpg_face_4.jpg	Low resolution
30359	closed_eye_0919.jpg_face_3.jpg	Partial open eyes
30374	closed_eye_0986.jpg_face_1.jpg	Low resolution (sunglass?)
30375	closed_eye_0987.jpg_face_1.jpg	One partial open eye
30382	closed_eye_1023.jpg_face_1.jpg	One open eye from another person
30386	closed_eye_1049.jpg_face_1.jpg	One open eye from another person
30389	closed_eye_1056.jpg_face_2.jpg	Low resolution
30395	closed_eye_1080.jpg_face_1.jpg	Low resolution
30399	closed_eye_1101.jpg_face_2.jpg	Partial open eyes
30401	closed_eye_1104.jpg_face_1.jpg	One partial open eye
30405	closed_eye_1113.jpg_face_3.jpg	Low resolution
30407	closed_eye_1117.jpg_face_1.jpg	Low resolution
30411	closed_eye_1126.jpg_face_1.jpg	Low resolution
30412	closed_eye_1127.jpg_face_1.jpg	Low resolution, 1 partial open eye
30424	closed_eye_1155.jpg_face_1.jpg	Partial open eyes
30439	closed_eye_1208.jpg_face_1.jpg	Low resolution, possible 1 open eye
30445	closed_eye_1229.jpg_face_1.jpg	One open eye from another person
30454	closed_eye_1244.jpg_face_1.jpg	Low resolution, possible 1 open eye
30455	closed_eye_1246.jpg_face_3.jpg	Low resolution
30457	closed_eye_1249.jpg_face_1.jpg	Low resolution
30458	closed_eye_1249.jpg_face_2.jpg	Partial open eyes
30459	closed_eye_1253.jpg_face_1.jpg	Possible 1 open eye
30463	closed_eye_1263.jpg_face_1.jpg	Reflection
30464	closed_eye_1264.jpg_face_2.jpg	Low resolution
30467	closed_eye_1267.jpg_face_1.jpg	One partial open eye
30468	closed_eye_1269.jpg_face_1.jpg	Partial open eyes

Table F.1 continued from previous page

Sample	Original file name	Observation
30470	closed_eye_1272.jpg_face_1.jpg	One partial open eye
30471	closed_eye_1274.jpg_face_1.jpg	Low resolution
30472	closed_eye_1276.jpg_face_2.jpg	Low resolution, possible 1 open eye
30473	closed_eye_1276.jpg_face_6.jpg	Partial open eyes
30474	closed_eye_1277.jpg_face_1.jpg	Low resolution, looking down
30475	closed_eye_1278.jpg_face_1.jpg	Low resolution, possible 1 open eye
30476	closed_eye_1278.jpg_face_3.jpg	Partial open eyes
30478	closed_eye_1281.jpg_face_1.jpg	Low resolution
30479	closed_eye_1282.jpg_face_1.jpg	Low resolution, possible 1 open eye
30484	closed_eye_1291.jpg_face_3.jpg	One partial open eye
30491	closed_eye_1301.jpg_face_2.jpg	One partial open eye
30495	closed_eye_1310.jpg_face_1.jpg	Low resolution, possible 1 open eye
30501	closed_eye_1319.jpg_face_1.jpg	One partial open eye
30503	closed_eye_1322.jpg_face_1.jpg	One open eye from another person
30505	closed_eye_1324.jpg_face_1.jpg	Partial open eyes
30506	closed_eye_1325.jpg_face_1.jpg	Low resolution
30509	closed_eye_1328.jpg_face_2.jpg	Low resolution, possible 1 open eye
30514	closed_eye_1333.jpg_face_10.jpg	low resolution
30517	closed_eye_1336.jpg_face_2.jpg	One open eye from another person
30518	closed_eye_1338.jpg_face_1.jpg	One open eye from another person
30520	closed_eye_1343.jpg_face_1.jpg	Partial open eyes
30521	closed_eye_1344.jpg_face_1.jpg	Low resolution
30525	closed_eye_1351.jpg_face_2.jpg	Low resolut., possible partial open eyes
30528	closed_eye_1356.jpg_face_10.jpg	partial open eyes
30534	closed_eye_1362.BMP_face_2.jpg	Low resolution, min. image size 73
30538	closed_eye_1363.jpg_face_1.jpg	Partial open eyes
30543	closed_eye_1376.jpg_face_2.jpg	Drawing
30547	closed_eye_1383.jpg_face_1.jpg	Low resolution
30550	closed_eye_1388.jpg_face_2.jpg	Low resolut., possible partial open eyes
30552	closed_eye_1394.jpg_face_1.jpg	Partial open eyes
30553	closed_eye_1394.jpg_face_3.jpg	Low resolution, min. image size 73
30554	closed_eye_1394.jpg_face_4.jpg	Low resolution
30557	closed_eye_1409.jpg_face_1.jpg	Low resolution
30558	closed_eye_1411.jpg_face_1.jpg	Low resolution?
30569	closed_eye_1445.jpg_face_1.jpg	Low resolut., possible partial open eyes
30573	closed_eye_1462.jpg_face_2.jpg	One partial open eye
30575	closed_eye_1464.jpg_face_1.jpg	Low resolution, min. image size 67
30577	closed_eye_1466.jpg_face_1.jpg	Low resolution
30583	closed_eye_1479.jpg_face_1.jpg	Low resolution
30584	closed_eye_1480.jpg_face_4.jpg	Low resolution
30585	closed_eye_1481.jpg_face_1.jpg	Partial open eyes
30586	closed_eye_1483.jpg_face_1.jpg	Low resolution, 1 partial open eye
30588	closed_eye_1486.jpg_face_2.jpg	Partial open eyes
30589	closed_eye_1486.jpg_face_4.jpg	Low resolut., possible partial open eyes
30590	closed_eye_1488.jpg_face_1.jpg	Low resolution, min. image size 65

Table F.1 continued from previous page

Sample	Original file name	Observation
30591	closed_eye_1489.jpg_face_1.jpg	Low resolution
30592	closed_eye_1489.jpg_face_2.jpg	Low resolution
30595	closed_eye_1495.jpg_face_1.jpg	One open eye from another person
30597	closed_eye_1497.jpg_face_1.jpg	Low resolut., possible partial open eyes
30603	closed_eye_1505.jpg_face_1.jpg	One open eye from another person
30605	closed_eye_1507.jpg_face_3.jpg	Low resolution
30606	closed_eye_1507.jpg_face_5.jpg	Low resolution
30607	closed_eye_1508.jpg_face_1.jpg	Low resolution
30608	closed_eye_1508.jpg_face_2.jpg	Low resolution
30609	closed_eye_1508.jpg_face_3.jpg	Low resolution
30613	closed_eye_1515.jpg_face_1.jpg	Low resolution
30619	closed_eye_1531.jpg_face_1.jpg	Partial open eyes
30621	closed_eye_1535.jpg_face_2.jpg	Partial open eyes
30622	closed_eye_1536.jpg_face_1.jpg	One partial open eye
30623	closed_eye_1537.jpg_face_1.jpg	Partial open eyes
30633	closed_eye_1556.jpg_face_3.jpg	Low resolution
30634	closed_eye_1558.jpg_face_1.jpg	Low resolution
30638	closed_eye_1567.jpg_face_1.jpg	Low resolution, min. image size 65
30640	closed_eye_1571.jpg_face_2.jpg	One partial open eye
30645	closed_eye_1580.jpg_face_1.jpg	Partial open eyes
30652	closed_eye_1594.jpg_face_1.jpg	One open eye from another person
30653	closed_eye_1600.jpg_face_1.jpg	Low resolution
30658	closed_eye_1618.jpg_face_2.jpg	Low resolution
30668	closed_eye_1641.jpg_face_1.jpg	Low resolution
30669	closed_eye_1641.jpg_face_2.jpg	Low resolution
30670	closed_eye_1641.jpg_face_3.jpg	Low resolution
30673	closed_eye_1648.jpg_face_2.jpg	Low resolution, min. image size 64
30681	closed_eye_1661.jpg_face_3.jpg	Partial open eyes
30684	closed_eye_1668.jpg_face_1.jpg	One partial open eye
30687	closed_eye_1671.jpg_face_1.jpg	Partial open eyes
30688	closed_eye_1673.jpg_face_2.jpg	Low resolution
30692	closed_eye_1680.jpg_face_1.jpg	Low resolution, 1 partial open eye
30693	closed_eye_1681.jpg_face_2.jpg	Partial open eyes
30696	closed_eye_1691.jpg_face_3.jpg	One partial open eye
30697	closed_eye_1692.jpg_face_1.jpg	Low resolution
30703	closed_eye_1702.BMP_face_1.jpg	Low resolut., possible partial open eyes
30704	closed_eye_1702.BMP_face_2.jpg	Low resolut., possible partial open eyes
30706	closed_eye_1707.jpg_face_2.jpg	Low resolut., possible partial open eyes
30708	closed_eye_1712.jpg_face_4.jpg	One open eye from another person
30709	closed_eye_1713.jpg_face_1.jpg	Low resolution, possible 1 open eye
30712	closed_eye_1722.jpg_face_3.jpg	UNSURE
30713	closed_eye_1730.jpg_face_1.jpg	Low resolution
30717	closed_eye_1742.jpg_face_1.jpg	Low resolution, possible 1 open eye
30724	closed_eye_1761.jpg_face_1.jpg	Low resolut., possible partial open eyes
30725	closed_eye_1762.jpg_face_1.jpg	Low resolution

Table F.1 continued from previous page

Sample	Original file name	Observation
30726	closed_eye_1763.jpg_face_2.jpg	Low resolution
30728	closed_eye_1768.jpg_face_1.jpg	Low resolution
30729	closed_eye_1769.jpg_face_2.jpg	Low resolution
30730	closed_eye_1770.jpg_face_1.jpg	Partial open eyes
30731	closed_eye_1773.jpg_face_1.jpg	Low resolution
30736	closed_eye_1782.jpg_face_3.jpg	One open eye from another person
30740	closed_eye_1789.jpg_face_1.jpg	Low resolution
30741	closed_eye_1796.jpg_face_1.jpg	Partial open eyes
30743	closed_eye_1804.jpg_face_1.jpg	Low resolution
30744	closed_eye_1805.jpg_face_1.jpg	Low resolution
30745	closed_eye_1806.jpg_face_1.jpg	Low resolut., possible partial open eyes
30746	closed_eye_1807.jpg_face_1.jpg	Low resolution
30748	closed_eye_1810.jpg_face_1.jpg	Low resolution
30754	closed_eye_1823.jpg_face_1.jpg	Low resolution
30755	closed_eye_1825.jpg_face_1.jpg	Partial open eyes
30761	closed_eye_1844.png_face_2.jpg	Low resolution
30765	closed_eye_1850.jpg_face_1.jpg	Low resolution, min. image size 65
30766	closed_eye_1850.jpg_face_2.jpg	Low resolut., possible partial open eyes
30767	closed_eye_1853.jpg_face_1.jpg	Low resolution
30770	closed_eye_1858.jpg_face_1.jpg	Partial open eyes
30771	closed_eye_1858.jpg_face_2.jpg	Partial open eyes
30772	closed_eye_1859.jpg_face_1.jpg	Low resolution
30775	closed_eye_1866.jpg_face_1.jpg	One open eye from another person
30776	closed_eye_1867.jpg_face_1.jpg	One partial open eye
30782	closed_eye_1879.jpg_face_2.jpg	Partial open eyes
30784	closed_eye_1884.jpg_face_2.jpg	Low resolution
30790	closed_eye_1894.jpg_face_1.jpg	UNSURE, looking down?
30793	closed_eye_1902.jpg_face_4.jpg	Low resolution, possible 1 open eye
30794	closed_eye_1903.jpg_face_1.jpg	Low resolution, possible 1 open eye
30795	closed_eye_1905.jpg_face_1.jpg	Low resolution
30797	closed_eye_1922.jpg_face_2.jpg	One open eye from another person
30802	closed_eye_1934.png_face_1.jpg	Low resolution
30803	closed_eye_1935.jpg_face_1.jpg	Low resolution
30808	closed_eye_1942.jpg_face_2.jpg	Low resolution
30810	closed_eye_1944.jpg_face_1.jpg	Low resolut., possible partial open eyes
30811	closed_eye_1946.jpg_face_2.jpg	Low resolut., possible partial open eyes
30813	closed_eye_1952.jpg_face_1.jpg	Low resolution
30818	closed_eye_1958.jpg_face_1.jpg	Partial open eyes
30823	closed_eye_1966.jpg_face_1.jpg	Low resolution, min. image size 70
30829	closed_eye_1982.jpg_face_1.jpg	Low resolution
30836	closed_eye_2005.jpg_face_2.jpg	Low resolution
30837	closed_eye_2008.jpg_face_1.jpg	Low resolution, possible 1 open eye
30838	closed_eye_2010.jpg_face_1.jpg	Low resolution
30842	closed_eye_2018.jpg_face_1.jpg	Image effects, not clear face contours
30843	closed_eye_2019.jpg_face_1.jpg	Low resolution

Table F.1 continued from previous page

Sample	Original file name	Observation
30844	closed_eye_2021.jpg_face_1.jpg	Low resolution
30845	closed_eye_2024.jpg_face_1.jpg	Low resolution, min. image size 71
30846	closed_eye_2025.jpg_face_2.jpg	Low resolution, partial open eyes
30850	closed_eye_2030.jpg_face_2.jpg	Low resolution
30851	closed_eye_2031.jpg_face_1.jpg	Low resolut., possible partial open eyes
30856	closed_eye_2038.jpg_face_2.jpg	Low resolution
30860	closed_eye_2046.jpg_face_1.jpg	Low resolution
30862	closed_eye_2049.jpg_face_1.jpg	Low resolution
30863	closed_eye_2050.jpg_face_1.jpg	Low resolution, min. image size 64
30867	closed_eye_2064.jpg_face_1.jpg	Low resolut., possible partial open eyes
30869	closed_eye_2071.jpg_face_1.jpg	Low resolution
30871	closed_eye_2075.jpg_face_1.jpg	Low resolut., possible partial open eyes
30872	closed_eye_2078.jpg_face_3.jpg	Low resolution, min. image size 74
30874	closed_eye_2087.jpg_face_2.jpg	Low resolution
30877	closed_eye_2094.jpg_face_1.jpg	Low resolution
30878	closed_eye_2095.jpg_face_1.jpg	Low resolution
30879	closed_eye_2097.jpg_face_1.jpg	One open eye from another person
30881	closed_eye_2099.jpg_face_1.jpg	Low resolution, possible 1 open eye
30886	closed_eye_2114.jpg_face_2.jpg	Low resolution, possible 1 open eye
30888	closed_eye_2118.jpg_face_1.jpg	One open eye from another person
30890	closed_eye_2123.jpg_face_2.jpg	Low resolution, possible 1 open eye
30893	closed_eye_2128.BMP_face_1.jpg	One partial open eye
30895	closed_eye_2132.jpg_face_3.jpg	One partial open eye
30896	closed_eye_2136.jpg_face_3.jpg	Partial open eyes
30902	closed_eye_2142.jpg_face_2.jpg	One partial open eye
30903	closed_eye_2144.jpg_face_2.jpg	Partial open eyes
30904	closed_eye_2145.jpg_face_4.jpg	Low resolution, possible 1 open eye
30905	closed_eye_2146.jpg_face_1.jpg	One partial open eye
30906	closed_eye_2147.jpg_face_2.jpg	Low resolution
30914	closed_eye_2164.jpg_face_1.jpg	Low resolution, possible 1 open eye
30915	closed_eye_2165.jpg_face_1.jpg	Partial open eyes
30921	closed_eye_2172.jpg_face_1.jpg	Partial open eyes
30922	closed_eye_2179.jpg_face_2.jpg	Partial open eyes
30926	closed_eye_2188.jpg_face_1.jpg	Partial open eyes
30931	closed_eye_2203.jpg_face_2.jpg	Partial open eyes
30932	closed_eye_2205.jpg_face_2.jpg	Image effects, not clear face features
30933	closed_eye_2207.jpg_face_1.jpg	Low resolution, min. image size 64
30936	closed_eye_2214.jpg_face_2.jpg	Low resolution, min. image size 72
30942	closed_eye_2222.jpg_face_1.jpg	Low resolution, min. image size 65
30945	closed_eye_2234.jpg_face_1.jpg	Low resolution
30946	closed_eye_2234.jpg_face_2.jpg	Low resolution
30947	closed_eye_2234.jpg_face_3.jpg	Low resolution
30948	closed_eye_2238.jpg_face_1.jpg	Low resolution
30950	closed_eye_2243.jpg_face_2.jpg	Partial open eyes
30952	closed_eye_2247.jpg_face_1.jpg	One partial open eye

Table F.1 continued from previous page

Sample	Original file name	Observation
30956	closed_eye_2251.jpg_face_1.jpg	Low resolution
30960	closed_eye_2259.jpg_face_1.jpg	Partial open eyes
30961	closed_eye_2260.jpg_face_2.jpg	Low resolution
30973	closed_eye_2304.jpg_face_16.jpg	low resolution
30977	closed_eye_2313.jpg_face_1.jpg	Partial open eyes
30989	closed_eye_2344.jpg_face_1.jpg	Low resolution, min. image size 59
30991	closed_eye_2352.jpg_face_1.jpg	Low resolution
30992	closed_eye_2352.jpg_face_2.jpg	Low resolution, 1 partial open eye
30999	closed_eye_2372.jpg_face_1.jpg	Partial open eyes
31000	closed_eye_2372.jpg_face_2.jpg	Low resolution
31001	closed_eye_2373.jpg_face_1.jpg	Low resolution
31004	closed_eye_2378.jpg_face_1.jpg	Partial open eyes
31006	closed_eye_2380.jpg_face_1.jpg	Low resolution
31014	closed_eye_2391.jpg_face_1.jpg	Image effects
31017	closed_eye_2395.jpg_face_1.jpg	Low resolut., possible partial open eyes
31021	closed_eye_2405.jpg_face_1.jpg	Low resolution, min. image size 66
31022	closed_eye_2406.jpg_face_1.jpg	Low resolution
31037	closed_eye_2446.jpg_face_1.jpg	Low resolution
31039	closed_eye_2448.jpg_face_1.jpg	Low resolution, min. image size 54
31046	closed_eye_2478.jpg_face_1.jpg	Low resolution
31047	closed_eye_2481.jpg_face_1.jpg	Low resolution, possible 1 open eye
31061	closed_eye_2506.jpg_face_1.jpg	Low resolution
31064	closed_eye_2511.jpg_face_1.jpg	Low resolution
31066	closed_eye_2513.jpg_face_2.jpg	Partial open eyes
31071	closed_eye_2526.BMP_face_2.jpg	One partial open eye
31074	closed_eye_2531.jpg_face_1.jpg	Low resolution
31082	closed_eye_2549.jpg_face_1.jpg	Low resolution
31083	closed_eye_2551.jpg_face_3.jpg	Low resolution
31084	closed_eye_2552.jpg_face_1.jpg	Low resolution
31085	closed_eye_2557.jpg_face_1.jpg	Low resolution
31086	closed_eye_2562.jpg_face_1.jpg	Drawing
31093	closed_eye_2582.jpg_face_1.jpg	Low resolution
31094	closed_eye_2585.jpg_face_1.jpg	Low resolution, min. image size 63
31095	closed_eye_2586.jpg_face_1.jpg	Low resolution, min. image size 74
31099	closed_eye_2601.jpg_face_1.jpg	Low resolution, min. image size 74
31101	closed_eye_2605.jpg_face_1.jpg	Low resolution
31102	closed_eye_2608.jpg_face_1.jpg	Low resolution, min. image size 65
31103	closed_eye_2612.jpg_face_1.jpg	Low resolution, min. image size 56
31105	closed_eye_2617.jpg_face_2.jpg	Low resolution, min. image size 70
31111	closed_eye_2631.jpg_face_1.jpg	Low resolution
31112	closed_eye_2634.jpg_face_1.jpg	Low resolution, min. image size 61
31115	closed_eye_2643.jpg_face_1.jpg	Low resolution
31118	closed_eye_2648.jpg_face_1.jpg	Partial open eyes
31119	closed_eye_2651.jpg_face_1.jpg	Low resolution
31120	closed_eye_2657.jpg_face_1.jpg	Low resolution

Table F.1 continued from previous page

Sample	**Original file name**	**Observation**
31126	closed_eye_2666.jpg_face_1.jpg	Low resolution, min. image size 65
31127	closed_eye_2670.jpg_face_1.jpg	Low resolution
31130	closed_eye_2680.jpg_face_2.jpg	Low resolution
31131	closed_eye_2681.jpg_face_2.jpg	Low resolution
31134	closed_eye_2689.jpg_face_4.jpg	Low resolution, min. image size 65
31154	closed_eye_2736.jpg_face_1.jpg	Drawing
31156	closed_eye_2738.jpg_face_1.jpg	Low resolution, min. image size 68
31162	closed_eye_2750.jpg_face_1.jpg	Low resolution
31163	closed_eye_2752.jpg_face_1.jpg	Low resolution, min. image size 70
31166	closed_eye_2757.BMP_face_1.jpg	One open eye from another person
31187	closed_eye_2804.jpg_face_1.jpg	Low resolution, min. image size 66

Source: Author.