university of groningen / faculty of science and engineering

# Improving face segmentation using feature extractors fine-tuned on gender recognition

Bachelor's Project Thesis

Arvid Lindström, s2740761, a.l.lindstrom@student.rug.nl,
Supervisors: Dr Marco Wiering

**Abstract:** This study investigates the benefit of fine-tuning encoders used in segmentation networks prior to training on semantic segmentation data. The domain of the study is face segmentation through pixel-wise labelling using three models: VGG16, VGG19, and ResNet-50 as encoders. A cross- comparison study is performed where encoders trained previously on Imagenet are compared with encoders trained on Imagenet followed by fine-tuning on a gender recognition task. The dataset used for gender recognition is the CelebA dataset. The datasets LFW and HELEN are used for face segmentation. It is demonstrated that segmentation networks built on VGG16 and VGG19 obtain an average IoU increase of 3.9% and 11.0% respectively when encoders are tuned on gender recognition prior to being used for face segmentation.

## 1 Introduction

Detection of faces in 2D images is a widely explored challenge in computer vision. It is a task with a large range of applications from surveillance, person recognition, and human-machine interaction. Since their inception, convolutional neural networks have demonstrated exceptional performance on a large range of computer vision tasks, outperforming previous hand-crafted feature models (Schmidhuber, 2014). Face detection is a problem particularly well suited for machine learning using convolutional neural nets. In recent years it has been demonstrated that the so-called *fully convolutional neural networks* can be trained to solve the problem of semantic segmentation (Long, Shelhamer, and Darrell, 2014). This task involves the pixel-wise prediction of classes in images and is an attractive choice for detecting faces. Unlike bounding-box proposals that can determine where a face appears using a rectangle, semantic segmentation allows for localization of faces with pixel-wise precision.

### 1.1 Contributions of this paper

This paper investigates how a fully convolutional neural network used for face segmentation benefits from fine-tuning on a related task prior to being trained for segmentation. It has been shown that attribute-aware networks can be constructed by training convolutional networks to perform some face-related classification task (Yang, Luo, Loy, and Tang, 2015). By stacking convolutional layers, object locations can be very roughly estimated by upsampling the output activations (Zeiler and Fergus, 2013). The present study leverages this technique in the pursuit of training semantic segmentation networks. The fully convolutional models in (Siam, Gamal, Abdel-Razek, Yogamani, and Jägersand, 2018) and (Long et al., 2014) use weights pretrained on the *Imagenet* dataset. The Imagenet dataset does not explicitly label "face" as a class. The aim of this study is therefore to investigate how a segmentation network improves from using weights pretrained on both Imagenet and a gender recognition task.
The models used for gender recognition fine-tuning are VGG16, VGG19 (Simonyan and Zisserman, 2014) and ResNet-50 (He, Zhang, Ren, and Sun, 2015a). The resulting convolutional layers are then used as encoders upon which three segmentation networks are constructed. The study proceeds by comparing the models' learning curves and performance, before and after the fine-tuning on gender data. A comparison is conducted across two bench-

mark datasets, LFW (Huang, Ramesh, Berg, and Learned-Miller, 2007) and HELEN (Vuong Le and Huang, 2012), and for each dataset the performance of a model is measured with and without hair being labelled as part of the face. The results obtained are used to compare the training times and validation IoU-scores across fine-tuned and non fine-tuned models.

### 1.2 Localizing objects using deep, upsampled convolutions

The paper which demonstrated the potential for face-attribute aware networks and subsequently inspired the work of this thesis is "From Facial Parts Responses to Face Detection: A Deep Learning Approach" by (Yang et al., 2015). Their study showed that by combining the upsampled activations from the deepest layers in multiple CNNs trained on facial parts recognition, a heatmap is created with localized responses from each CNN. The convolutional networks in their study were based on AlexNet (Krizhevsky, Sutskever, and Hinton, 2012) and were trained, respectively, on tasks such as detecting the type of hair of a subject or the type of mouth. In this study we combine this method with transposed convolutions in order to perform pixel-wise labelling of faces using a "gender-aware" encoder. To the best of our knowledge, no research has been done measuring how a segmentation network performs given domain-aware encoders. In most literature, the task of a segmentation network is to label pixels of a large amount of classes, such as in (Long et al., 2014). Other work in the literature attempts to benchmark the computational efficiency of fully convolutional networks (Siam et al., 2018).

### 1.3 Outline

Section 2 describes the previous work done in the field relevant to this study. The models and experiments are described in sections 3 and 4 respectively. The results of the study are presented in section 5. Section 6 concludes this paper and describes possibilities for future work.

## 2 Previous Work

### 2.1 Convolutional neural networks

Convolutional neural networks used for feature extraction have been overwhelmingly successful in the field of computer vision since their original introduction in (Lecun, Bottou, Bengio, and Haffner, 1998). Convnets constitute a large part of modern machine learning and over the years many architectures based on convolutional layers have been designed to solve difficult detection and classification problems. Ciresan et al. were one of the first to show the potential of training deep nets using graphical processing units (GPUs) (Ciresan, Meier, Gambardella, and Schmidhuber, 2010). In 2012, Krizhevsky et al., (2012) revolutionized the field by winning the 2012 ILSVRC competition using a deep network trained on GPUs using the dropout regularization method. These important studies set the ground for the deeper models VGG16/19 (Simonyan and Zisserman, 2014) and ResNet-50 (He et al., 2015a) used in this paper. It has been demonstrated that deep Convnets can achieve human performance in recognizing faces (Taigman, Yang, Ranzato, and Wolf, 2014) and that Convnets can achieve state of the art performance on gender recognition under large pose-variation (v. d. Wolfshaar, Karaaba, and Wiering, 2015).

### 2.2 Fully convolutional neural networks

Previous methods of performing semantic segmentation used so-called "patch classification", in which each pixel would have its class predicted given an image of the surrounding pixels (Ciresan, Giusti, Gambardella, and Schmidhuber, 2012). This method, however, requires images of a fixed size due to the fully connected layers in the classifier. In 2014, (Long et al., 2014) showed that neural networks consisting solely of convolutional layers can exceed the previous state of the art performance on the PASCAL VOC dataset with a mean IoU score of 62.2%. Fully convolutional neural networks use *transposed convolutions*, also known as deconvolution. By transforming the max-pooling outputs from feature to label space using $1 \times 1$ convolutions as explained in (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, and

Rabinovich, 2014), a mapping is created where a max-pooling layer with dimensions $W \times H \times Depth$ is transformed into $W \times H \times K$, where $K$ is the number of classes. At this stage, transposed convolution takes place which can be seen as a *backwards strided convolution* using trained kernels (Long et al., 2014). The output of such a network will be the same size as the input image and with a depth equal to $K$.

## 2.3 Face segmentation in images

Smith and Yang (2013) and Nirkin, Masi, Tran, Hassner, and Medioni (2017) have explored face segmentation. Smith and Yang (2013) proposed an algorithm using a database of exemplar-based face images with corresponding segmentation masks. The algorithm determines the probability of a pixel belonging to a facial parts label, such as mouth, eyes or nose, by performing comparisons between other aligned face masks in the exemplary dataset. The largest contribution of (Smith and Yang, 2013) was the extension of the HELEN dataset (Vuong Le and Huang, 2012) by providing pixel-wise labels for future work as well as showing that segmentation masks are at least as informative as the previous standard of using landmark notations for mouths, eyes etc.
In (Nirkin et al., 2017), the authors demonstrated that a fully convolutional neural network, namely the FCN8s architecture from (Long et al., 2014), could outperform previous handcrafted methods given a rich enough dataset of training images. By picking out faces from the IARPA Janus CS2 dataset they managed to provide their network with 9818 training images, a considerable increase from previous benchmark datasets (Huang et al., 2007), (Vuong Le and Huang, 2012).

# 3 Architectures

In this study, the encoder-decoder architecture is adopted to provide means of quantitatively researching the effects of gender-aware feature extractors. Three models are considered as the encoder parts of the networks and a singular decoder is used for a direct comparison. The encoders receive an image as input and produce pooling activations which are later upsampled by the decoder and used to perform pixel-wise labelling of the image.

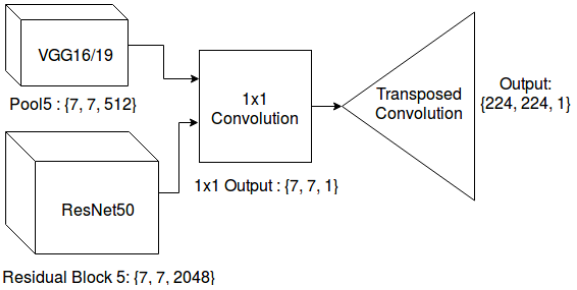



**Figure 3.1: A comparison between the dimensions of the VGG16/19 decoder and the ResNet-5050 decoder.**

## 3.1 The segmentation decoder

The decoder used in this paper is based on the FCN32 architecture devised by Long et al. (2014). Like the FCN32 model, only the final layer of the encoder is used to upsample the features into segmented-image space. The layers of the decoder can be seen in figure 3.1. The decision not to include the skip-connections proposed by (Long et al., 2014) was to avoid the need to tune the corresponding layers of the encoders during gender classification. As such the resulting decoder does not have access to the fine-grained features of the architectures FCN16s and FCN8s.
Once the activations from the encoders are produced (with respective output dimensions as seen in figure 3.1), a $1 \times 1$ convolution takes place. This is a learned convolution which transforms the input from feature space to label space. This convolution does not use a neuron activation function and the kernel weights are initialized using *He normal* initialization (He, Zhang, Ren, and Sun, 2015b) and L2 regularization to avoid overfitting (Cortes, Mohri, and Rostamizadeh, 2012).
The transposed convolution, as explained by (Long et al., 2014), uses a singular filter with kernel size $k = \{64, 64\}$ and stride $s = 32$ (hence the name FCN32). This layer uses a sigmoid activation to map each pixel value to $\{0 : \text{background}, 1 : \text{face}\}$. The weights in the transposed convolution layer are initialized to perform bilinear interpolation. This entails that without training the transposed convolution layer will initially magnify the output

from the $1 \times 1$ convolution into an image of $7 \times 7$ squares with a total dimension of $224 \times 224$. The motivation behind using such a weight initialization is to reduce training time by initially telling the decoder to upsample the accumulated pooling activations resulting from the $1 \times 1$ convolution.

## 3.2 The gender classifier

Using the feature extractors from the architectures VGG16, VGG19 and ResNet-50, a gender classifier is implemented with the intention of fine-tuning the final convolutional layer and the residual block of VGG16/19 and ResNet-50 respectively. The classifier is built as an MLP with the following layers:
*Flatten*: The first layer receives as input the final max-pooling layer of VGG16/19 and in the case of ResNet-50, the final average pooling layer of the 5th residual block. The dimension of the output is flattened from {7, 7, 512} to {25088} for VGG16/19 and from {1, 1, 2048} to {2048} for ResNet-50.
*Fully Connected 1*: A fully connected layer with 10 neurons (5 for VGG16) using the ReLU activation function (Nair and Hinton, 2010).
*Fully Connected 2*: A fully connected layer with 5 neurons using a sigmoid activation function.
*Softmax*: Two output neurons signify the classes *Male* and *Female* represented as 1 and 0 respectively. Using softmax, the output of the network can be interpreted as a probability distribution across the two classes. The parameters for the gender classifier were chosen through experimental trials with the aim of keeping the amount of parameters in the MLP as low as possible.

The motivation for the final parameter settings is to enforce as much adaptation on the feature extractors as possible during training on gender recognition. If the amount of parameters in the MLP is too high the network will be able to achieve high accuracy without discovering useful features in the final convolutional layers.
During training, all layers in the respective architectures are "frozen" except for the final convolutional blocks of VGG16/19 and the final residual block of ResNet-50. This entails that error information only propagates through the MLP and the final, unfrozen, blocks of the feature extractors. The amount of trainable parameters in these final convolutional blocks can be seen in table 3.1 along with the parameters of the MLP for gender classification.

**Table 3.1: Number of trainable parameters in each layer of the gender classifiers.**

| | **Architecture** | | |
|---|---|---|---|
| **Layer** | **VGG16** | **VGG19** | **ResNet-50** |
| **Conv.** | 7,079,424 | 9,439,232 | 14,976,000 |
| **FC-1** | 125,445 | 250,890 | 20,490 |
| **FC-2** | 30 | 55 | 55 |
| **Softmax** | 12 | 12 | 12 |

## 3.3 VGG16

The first architecture considered as an encoder for the segmentation network is VGG16 (Simonyan and Zisserman, 2014). This architecture was proposed in 2014 and has since been used as a benchmark encoder network for semantic segmentation (Long et al., 2014), (Siam et al., 2018). VGG16 was originally proposed by (Simonyan and Zisserman, 2014) as an extension of the work done by (Krizhevsky et al., 2012). It proposes the notion of using several $3 \times 3$ convolutions after another to mimic the effect of a singular, larger kernel such as the ones used in (Krizhevsky et al., 2012). In (Long et al., 2014), this network, among others, was reconstructed into a fully convolutional neural network and achieved state of the art performance with a mean IoU score of 56.0 on the PASCAL VOC 2011 dataset (Everingham, Van Gool, Williams, Winn, and Zisserman) with 21 classes.
The process of converting VGG16 to a fully convolutional network was to discard the final classification layer (with 1000 outputs) and converting the two hidden layers of 4096 neurons each to convolutional kernels with size $7 \times 7$ and stride $s = 0$. Intuitively this can be seen as using convolutions where each feature map behaves as a singular hidden neuron fully connected to its input (which in the case of VGG16/19 and ResNet-50 is $7 \times 7$ in size from the final pooling layer). This network, together with the encoder described in section 3.1, has approximately 134 million parameters.
In this study, the resulting FCN32s network has been further reconstructed by discarding the final convolutional layers and applying $1 \times 1$ convolution directly to the final pooling layer of VGG16, followed by transposed convolution. This is a design

choice intended to leverage the final convolution layer of VGG16 rather than the one proposed by (Long et al., 2014) as to allow a direct comparison between the gender-tuned versus non-gender-tuned convolutional layers. Following this reconstruction of the FCN32 architecture, the amount of parameters is equal to the ones in the VGG16 feature extractor ($\approx$ 14 million) with the addition of the $1 \times 1$ convolution and the transposed convolution ($513 + 4096$). This vast parameter reduction from 134 million to roughly 14.7 million is more suitable for a segmentation task of frontalized faces as compared to the previous parameter size used for 21 classes in unconstrained environments.

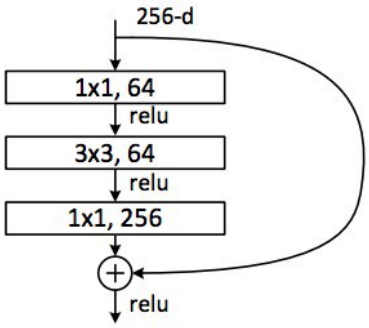



**Figure 3.2: Bottleneck layer of ResNet-50/101/152 (He et al., 2015a).**

### 3.4 VGG19

The VGG19 architecture (Simonyan and Zisserman, 2014) is chosen as the second architecture for this study due to its similarity with VGG16 and the increased amount of learnable parameters in its final convolutional block. The amount of parameters tuned during gender recognition is 9.4 million, as compared to the 7 million of the final convolutional block of VGG16. It should be addressed that the authors of (Long et al., 2014) did not observe improvements of using VGG19 as an encoder compared to VGG16, however since the segmentation tasks of this study and the aforementioned work differ we believe VGG19 is worth revisiting. The method for reconstructing VGG19 into an encoder-decoder network used for semantic segmentation is identical to the one described in the previous section.

### 3.5 ResNet-50

ResNet-50 (He et al., 2015a) is the 50 layers deep version of the ResNet architecture. Along with VGG16/19, this architecture is used in this study as an encoder for a face segmentation network. The ResNet-50 architecture utilizes skip connections to feed forward the input to convolutional block$_i$ directly to convolutional block$_{i+1}$. This entails that in the case where convolutional block$_i$'s optimal solution is to approximate the identity function, the weights of block$_i$ may be driven to 0 and block$_{i+1}$ will receive the same input as block$_i$. The ResNet-50 architecture utilizes this technique in conjunction with bottleneck layers introduced by (Szegedy et al., 2014). Figure 3.2 illustrates how ResNet-50 performs $3 \times 3$ convolutions on a downsampled set of features, drastically reducing computational cost. The skip connection depicted illustrates how ResNet-50 (and its deeper siblings) can grow very large without suffering from the vanishing gradient problem.

In this study, all layers of ResNet-50 except for the fifth (and final) residual block are frozen during tuning on gender recognition. The final layer of the fifth residual block is followed by an average pooling operation which is used as input to the decoder when constructing the segmentation network. Figure 3.1 shows the output dimensions of ResNet-50 in relation to the decoder described in section 3.1.

## 4 Experimental Setup

The execution of this study utilized the Keras python framework (Chollet, 2015), using Tensorflow as a backend (Abadi and Agarwal, 2015).

### 4.1 The CelebA dataset

The dataset CelebA (Liu, Luo, Wang, and Tang, 2015) is used for training and evaluating the gender classifiers. From the CelebA dataset consisting of over 200,000 images, a subset of 12,000 images are extracted. 10,800 images are used for training the gender classifiers and 1200 images are used for validation. The dataset contains images of celebrities with a rich set of backgrounds and facial-pose variations. Out of the 40 labeled attributes available to each image, only the gender class is used.

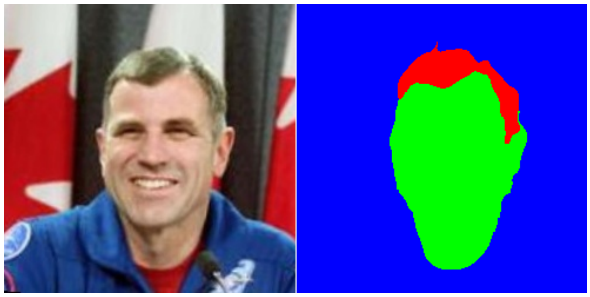


**Figure 4.1: Image of Dave Williams with the mask from LFW (Huang et al., 2007).**

## 4.2 The LFW dataset

The first dataset used for training and validation of the face-segmentation models is a subset of the *Labelled Faces in the Wild* dataset (Huang et al., 2007) with corresponding pixel-wise label annotations. This dataset contains 2927 images, out of which 2634 images (90%) are used for training and 293 images are used for validation. This dataset contains initially 3 classes for each label mask: background, hair, and face. The *hair*-class also includes facial hair. See Figure 4.1 for an example of an input image and its corresponding label mask. All images of LFW have been resized to 224 by 224 pixels.

## 4.3 The HELEN dataset

The second dataset used for training and validation is a subset of the HELEN dataset (Le and Huang, 2012) of facial landmarks (Smith and Yang, 2013). This dataset is used to increase the robustness of this study and to strengthen the power of the experimental results. The full dataset of segmented images consists of 2330 images from which 2097 (90%) of the images are used for training and 233 images are used for validation. Due to the nature of the dataset, some pre-processing was necessary to fully utilize the provided labels. The original dataset labelled facial parts against the background and provided 11 label masks for each input image. These masks contained separate labels for each eye, the nose, the hair and seven other designated facial parts. Each label mask provides each pixel a continuous label from 0.0 to 1.0 representing the degree to which a pixel belonged to either, for example a nose or a cheek. For this study, each pixel which has an agreement with a facial part of at least 0.5, as compared to being background, was set to 1.0, indicating that the pixel should be labelled as part of a face. Just as with the LFW dataset, all images used have been resized to 224 by 224 pixels.

## 4.4 Optimizers and loss functions

When training the gender classifiers and the face-segmentation models, binary cross entropy is used as the loss function:

$$-y \times \log(p) - (1 - y) \times \log(1 - p) \tag{4.1}$$

For segmentation, this loss function can be seen as a two-dimensional grid of single, binary classifications for classes *face* and *non-face* for each pixel. Other loss-functions have been proposed in the literature such as the softmax- cross entropy loss in (Long et al., 2014) for multiple classes or the work of (Atiqur Rahman and Wang, 2016) which demonstrates that IoU-loss can be directly optimized and differentiable for segmentation. This study advocates the use of sigmoid activation functions and binary cross entropy loss to increase the comparative validity of the research.
The optimizer used is the RMSprop optimizer proposed in a Coursera course from 2012 (Tieleman and Hinton, 2012). The purpose of this optimizer is to keep a running average of squared gradients used so far and to divide the current gradient update with the current average as to ensure that all weights are updated more or less equally. For this study, the default recommended values for the decay rate was set to $\lambda = 0.9$ whereas the learning rate $\alpha$ was set to 0.0001 through trial experiments. It was found that using stochastic gradient descent through tweaking of parameters based on the ones used by (Long et al., 2014) did not achieve comparable learning results across different model types. RMSprop appears to behave more consistently across different training sessions and was therefore selected as the optimizing function.

## 4.5 Models for gender recognition

The first stage of this study is to train gender classifiers using the dataset CelebA (Liu et al., 2015) and the architectures VGG16/19 and ResNet-50. The models are constructed as described in section

3.2 and trained using a batch size of 200 images. Each gender classifier was set to terminate its training if the model did not show a decrease in the loss on the validation data for longer than 20 epochs. During training, the weights currently performing best (highest accuracy) on the validation data are saved. The results of the gender recognition training on CelebA can be seen in table 4.1. The epoch listed in the 2nd column refers to the last epoch in which the model improved on validation data before training was terminated. It can be seen that with the least amount of epochs, VGG16 outperforms the other two models with an accuracy of 97.5%. All models perform well on the classification task and further comments on potential improvements of this training are discussed in section 6.2.

### 4.6 Models for face segmentation

Training of the segmentation models was divided into eight distinct experimental conditions. The steps for each model (VGG16, VGG19 and ResNet-50) are as follows:

1. Construct the model into a segmentation network following the method in section 3.1.
2. Freeze all layers in the encoder.
3. Train the control model without using gender tuned weights:
   (a) Train on the LFW-dataset while including hair in the face label-mask
   (b) Train on the LFW-dataset without including hair in the face label-mask
   (c) Train on the HELEN-dataset while including hair in the face label-mask
   (d) Train on the HELEN-dataset without including hair in the face label-mask
4. Record the segmentation output, IoU-score and training curves for each session.
5. Train a gender-tuned model by initializing the encoder with weights from gender training on CelebA
6. Repeat step 2-4 using gender-tuned model.

All segmentation models were trained under the conditions that training could take no longer than 20 hours on a single NVIDIA k40 GPU, using a batch size of 200. Training was stopped when the validation IoU-scores did not increase over 50 epochs. The decision to freeze all layers of the encoder during segmentation training was made to ensure that the influence of gender-tuned encoders could be directly compared to the control models. Allowing gradient updates from error on LFW/HELEN-data was shown to eventually have such a large influence on the encoders that the final difference in performance became indistinguishable. This occurred after a relatively large amount of epochs making only the first few epochs comparable in terms of model performance.

**Table 4.1: Results for gender recognition on validation data**

| Architecture | Epochs trained | Accuracy |
|---|---|---|
| **VGG16** | 23 | 97.50% |
| **VGG19** | 26 | 96.58% |
| **ResNet-50** | 32 | 94.42% |

## 5 Results

In this section, the models trained without prior gender tuning are referred to as *control models* and the models trained with prior gender tuning as *gender models*.

### 5.1 Intersection over union scores

This study uses the validation data to report IoU-scores. For each epoch of training, the IoU score is calculated on the validation set of the LFW and HELEN datasets as a mean IoU across all predictions and corresponding labels of the validation data.

**Table 5.1: Validation IoU scores for VGG16**

| Model | LFW | | HELEN | |
|---|---|---|---|---|
| | Hair | No hair | Hair | No hair |
| **Control** | 0.638 | 0.588 | 0.534 | 0.552 |
| | Hair | No hair | Hair | No hair |
| **Gender** | 0.660 | 0.625 | 0.542 | 0.578 |

Table 5.1 shows the maximum achieved IoU scores under the training constraints presented in section 4.6 for VGG16. Tables 5.1/2/3 contain the maximum IoU scores for all models. By maximum IoU we mean the IoU score on validation data at the final epoch in which the model improved on the validation set. It is important to note that the values presented in tables 5.1/2/3 have been rounded to the nearest 3rd decimal point for readability. For both datasets and for both conditions of inclusion of hair, the IoU score is higher for the gender model as compared to the control model with an average increase of 3.9%.

The same trend is observed in Table 5.2 for the VGG19 encoder. All IoU validation scores have increased with an average of 11.0%. The reported average improvements in IoU were calculated by computing, for each model, the difference in IoU scores between gender models and control models (as reported in tables 5.1/2/3) and averaging the results by dividing the sum of differences by 4 (for each experimental condition).

The results of using ResNet-50 as an encoder can be seen in Table 5.3. Clearly, there are no significant changes in IoU across the control model and the gender model. A more detailed analysis follows in section 6.

## 5.2 Comparison of training epochs

### 5.2.1 VGG16

The training curves for the segmentation network using VGG16 as encoder can be seen in Fig. 5.1 and

**Table 5.2: Validation IoU scores for VGG19**

| Model | LFW | | HELEN | |
|---|---|---|---|---|
| | Hair | No hair | Hair | No hair |
| **Control** | 0.605 | 0.538 | 0.514 | 0.515 |
| | Hair | No hair | Hair | No hair |
| **Gender** | 0.663 | 0.626 | 0.547 | 0.575 |

**Table 5.3: Validation IoU scores for ResNet-50**

| Model | LFW | | HELEN | |
|---|---|---|---|---|
| | Hair | No hair | Hair | No hair |
| **Control** | 0.671 | 0.604 | 0.581 | 0.587 |
| | Hair | No hair | Hair | No hair |
| **Gender** | 0.671 | 0.604 | 0.581 | 0.587 |

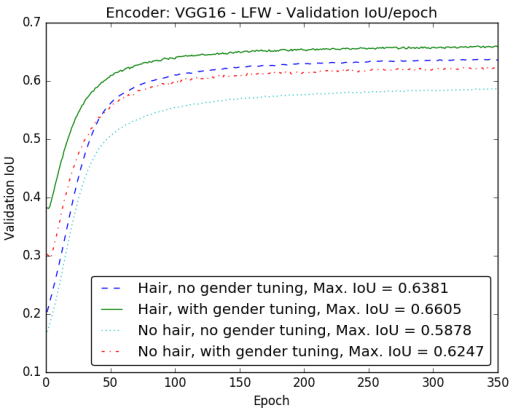



**Figure 5.1: Validation IoU over epochs for VGG16 encoder using LFW**

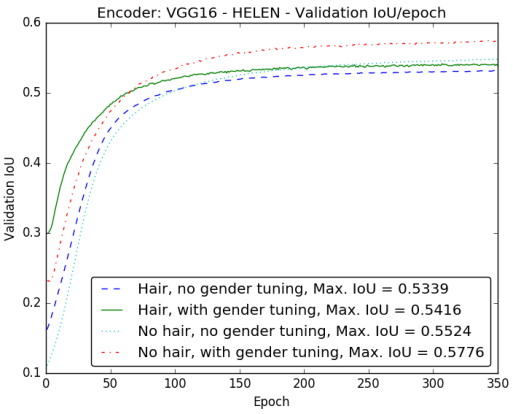



**Figure 5.2: Validation IoU over epochs for VGG16 encoder using HELEN**

Fig. 5.2. It can be seen that the HELEN dataset has proved to be more challenging for our model to segment. For the LFW-dataset, images containing hair have been easier to segment compared to images without hair, the opposite observation can be made for the HELEN dataset. Overall, on both datasets, the VGG16 based model demonstrates a head-start in validation accuracy for the gender model. The slope of the IoU increase over epochs is comparative across all conditions. However, the gender model reaches, and surpasses, the peak of the control model roughly 250-300 epochs earlier.

### 5.2.2 VGG19

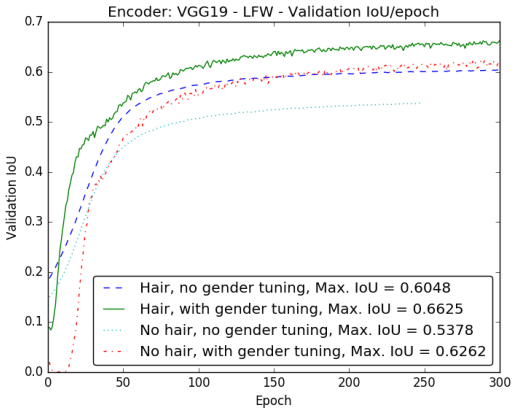



**Figure 5.3: Validation IoU over epochs for VGG19 encoder using LFW**

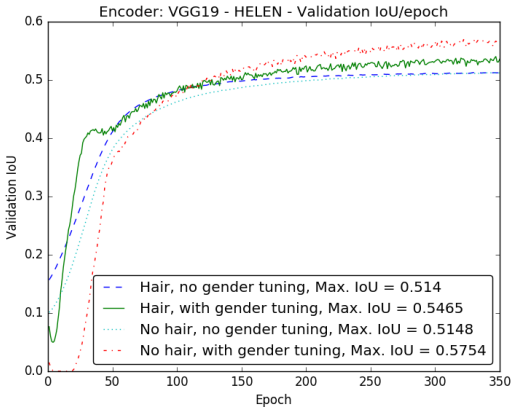



**Figure 5.4: Validation IoU over epochs for VGG19 encoder using HELEN**

Looking at Fig. 5.3, the amount of training epochs required to reach peak IoU with the control model is comparable with the results of VGG16. The VGG19 model does not however demonstrate the head start observed with the VGG16 models. The validation IoU over epochs fluctuates for the gender models until a smoother curve can be observed at around 50 epochs. This indicates that the gradient updates are becoming more stable and the gender models, under respective conditions, begin to slowly climb to their eventual peak IoU-scores. *Note:* the abrupt stop of the *No hair, no gender tuning* curve in Fig. 5.3 is due to the model no longer improving past that point, as such, training was terminated. The corresponding IoU score in table 5.2 has been rounded to the nearest 3rd decimal point. Overall the VGG19 model appears to perform at least as good as VGG16 across all experimental conditions and datasets.

## 5.3 ResNet-50

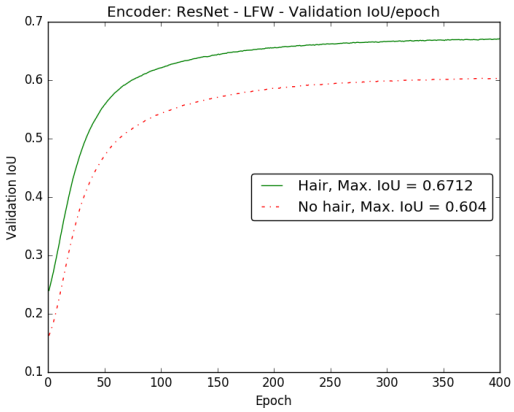



**Figure 5.5: Validation IoU over epochs for ResNet-50 encoder using LFW**

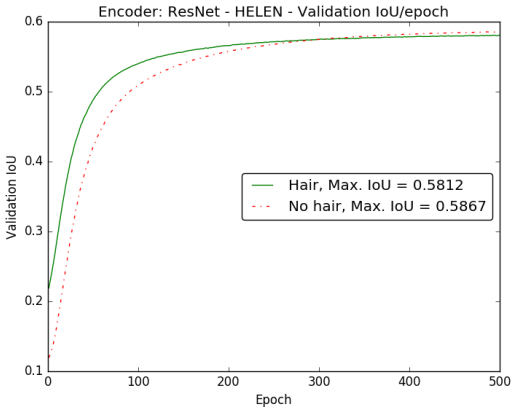
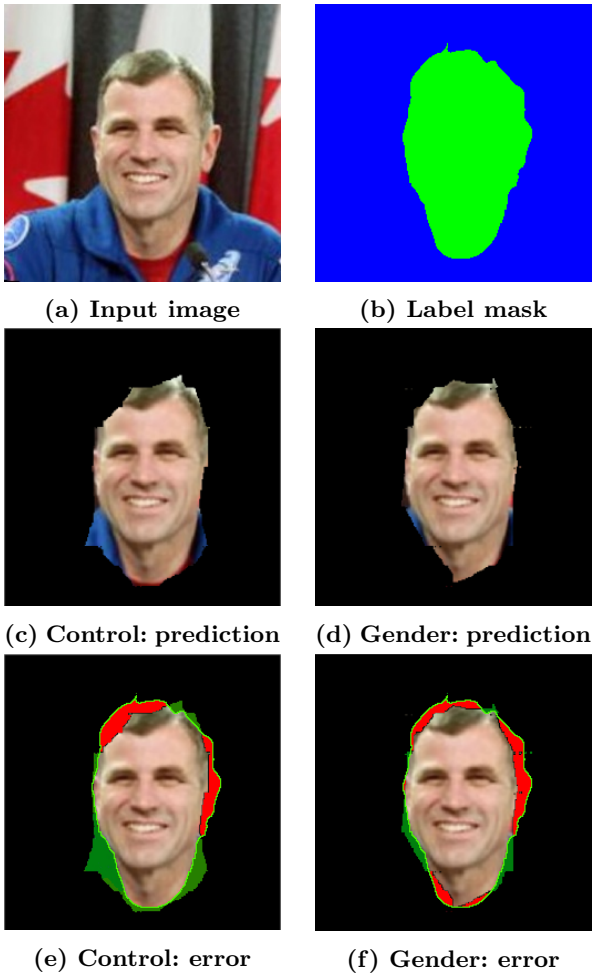



**Figure 5.6: Validation IoU over epochs for ResNet-50 encoder using HELEN**

As can be seen in Fig. 5.5 and in Fig. 5.6, the models using ResNet-50 as encoder neither improves or decreases in IoU-score on the validation

set. The learning curves are in fact virtually identical and have for visibility been plotted as a singular line for control model and gender model.

### 5.4 Segmentation results



**(a) Input image** **(b) Label mask**

**(c) Control: prediction** **(d) Gender: prediction**

**(e) Control: error** **(f) Gender: error**

**Figure 5.7: Face segmentation using VGG16 trained on LFW including Hair (best viewed in color), validation IoU of 66.0%.**

To demonstrate the validity of the resulting models, predictions are plotted as seen in Fig. 5.7. In Fig. 5.7e and Fig. 5.7f the false positives have been highlighted in green and the false negatives in red. It can be seen that for this particular example, the gender tuned model has noticeably decreased in false positives, however the ear and parts of the hair of the subject have been incorrectly labeled as non-face.

## 6 Conclusion and Future work

### 6.1 Interpretation of results

This study compared the performance and training times of face segmentation networks fine-tuned to perform gender recognition using the CelebA dataset (Liu et al., 2015) prior to being trained to segment faces using the LFW (Huang et al., 2007) and HELEN (Vuong Le and Huang, 2012) datasets. The segmentation networks were built upon the encoders (feature extractors) of the architectures VGG16, VGG19 (Simonyan and Zisserman, 2014) and ResNet-50 (He et al., 2015a). IoU-scores on validation data from the aforementioned datasets and training epochs were compared across models previously trained on the Imagenet dataset and models trained on Imagenet followed by fine-tuning on gender recognition. The results showed that tuning on gender recognition on a separate dataset gives a VGG16-based face segmentation network a "head-start" during training. In the case of a VGG19 based model, the gender-tuned model initially starts off worse but displays a steeper increase in IoU over epochs and thus learns faster (and better) than its corresponding control model. The model using ResNet-50 as an encoder does not display faster learning times or an increase in IoU. The performance has neither increased or decreased. We believe that the vast amount of parameters in the final residual block of ResNet-50 (approximately 15 million) requires much more than 10800 training images on a gender-recognition task to fully develop meaningful features.

In summary, the prior training on gender recognition benefits models VGG16 and VGG19 by increasing their validation IoU scores on LFW and HELEN as well as reducing training epochs. On ResNet-50 no change is observed and a more robust experiment using more data is required to draw any substantial conclusions on ResNet-50.

It is clear that the LFW dataset is more easily segmented when hair is included in the label mask. When hair is not included the IoU scores of all models is reduced for LFW. This trend is effectively reversed for the HELEN dataset. It was hypothesized that, due to hair being a naturally dividing feature

between males and females, the inclusion of hair in the labels would lead to a much greater improvement in segmentation for gender tuned models. The results are however not indicative of such a trend.

### 6.2 Future work

The method proposed in this paper does not tune layers in the encoder during training on segmentation data. This results in lower IoU scores, although the model is still capable of classifying faces in an image to a satisfying degree, see figure 5.7. The amount of learnable parameters for the network has also been significantly decreased for a direct comparison. It can be easily demonstrated that a model using a decoder with more parameters, such as FCN32 (Long et al., 2014), will achieve a higher IoU score on the LFW and HELEN datasets. However, this greatly diminishes the benefit of gender-tuning and makes comparisons difficult. A natural extension of this research is as follows:

- More challenging segmentation dataset: LFW's subset of segmented faces are mostly centered in the image and face towards the camera. This makes it easy to train models that perform well on the dataset and the benefit of gender-aware encoders is harder to investigate.
- Increased difficulty of gender recognition task: This paper only used 12000 images from the 202,000 images CelebA (Liu et al., 2015) due to computational constraints. As of such learning was done very quickly and the complexity of the encoders did not reflect the complexity of the gender recognition task (as evident from table 4.1).
- Given the first mentioned extension, a more sophisticated decoder is required to investigate if the current state of the art performance can be extended using gender-tuning. Using the decoder from FCN32 proposed by (Long et al., 2014), it is possible to achieve IoU scores greater than 0.9 within only a few epochs, however, as previously mentioned, this will make a comparison across differently tuned encoders hard to perform.
- Using facial parts aware encoders: This is a proposition to extend the work by (Yang et al., 2015) by concatenating the convolutional outputs of smaller encoders trained to classify facial parts such as the type of nose a subject has or the style of hair. By combining the outputs of many such networks and feeding the activations into a deconvolution network, it is hypothesized that a face segmentation network should display a vaster improvement as compared to the binary gender recognition tuning presented in this study.

## 7 Acknowledgements

The author would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster. A large thank you is also dedicated to Dr. Marco Wiering for the guidance and support leading up to this paper.

## References


Martín Abadi and Ashish Agarwal. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *Advances in Visual Computing: 12th International Symposium*, volume 10072, pages 234–244. December 2016. ISBN 978-3-319-50834-4.

Franois Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

Dan Ciresan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images.pdf`.

Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. *CoRR*, abs/1003.0358, 2010. URL `http://arxiv.org/abs/1003.0358`.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. *CoRR*, abs/1205.2653, 2012. URL `http://arxiv.org/abs/1205.2653`.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. URL `http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015a. URL `http://arxiv.org/abs/1512.03385`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015b. URL `http://arxiv.org/abs/1502.01852`.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. URL `http://arxiv.org/abs/1411.4038`.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL `http://dl.acm.org/citation.cfm?id=3104322.3104425`.

Yuval Nirkin, Iacopo Masi, Anh Tuan Tran, Tal Hassner, and Gérard G. Medioni. On face segmentation, face swapping, and face perception.

*CoRR*, abs/1704.06729, 2017. URL http://arxiv.org/abs/1704.06729.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828, 2014. URL http://arxiv.org/abs/1404.7828.

Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, and Martin Jägersand. RTSeg: Real-time semantic segmentation comparative study. *CoRR*, abs/1803.02758, 2018. URL http://arxiv.org/abs/1803.02758.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Jonathan Brandt Zhe Lin Smith, Li Zhang and Jianchao Yang. Exemplar-based face parsing. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Adobe Research, 2013. URL http://www.cs.wisc.edu/~lizhang/projects/face-parsing/.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014. doi: 10.1109/CVPR.2014.220.

T. Tieleman and G. Hinton. Lecture 6.5 - RMSprop, 2012.

J. v. d. Wolfshaar, M. F. Karaaba, and M. A. Wiering. Deep convolutional neural networks and support vector machines for gender recognition. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 188–195, Dec 2015. doi: 10.1109/SSCI.2015.37.

Zhe Lin Lubomir Boudev Vuong Le, Jonathan Brandt and Thomas. S. Huang. Interactive facial feature localization. *12th European Conference on Computer Vision*, ECCV, 2012. URL http://www.ifp.illinois.edu/~vuongle2/helen/.

Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. *CoRR*, abs/1509.06451, 2015. URL http://arxiv.org/abs/1509.06451.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.