In the Wild Face Parsing in Synthetic-Data Trained Models

Nathan Baker

Abstract

The benefits of synthesizing training data for deep learning have been enjoyed by the computer vision community for years. Face parsing, however, has seen little research in this area and a large domain gap has historically existed between real and synthetically generated faces for machine learning. It is only until 2021 that Microsoft released the first fully synthetic face dataset, allowing for research to be conducted on the viability of synthetic face data for computer vision tasks. Until now, no further research has been done regarding Microsoft's face dataset, and although their results were promising, it left a lot to be determined about the state of face parsing in the wild. In this paper, I will explore the effectiveness of computer-synthesized training data for face parsing, and answer the question if synthetic data is now at the point at which it can compete or even outperform real training data in the task of face parsing in the wild.

***Keywords*—** Face Parsing, Synthetic Training Data, in the wild, Domain Gap, Domain Adaptation, Semantic Segmentation

DECLARATION

I certify that all material in this dissertation which is not my own work has been identified.

_____ _____

Signature Date

Contents

1 Introduction

The task of face parsing involves pixel-wise segmentation of different semantic components (e.g. eyes, nose, mouth) from a facial image (see Figure: 15). Its importance can be understood from its use cases such as human emotion recognition [1], virtual beauty and makeup [2], and face identification [3]. And as in most deep learning problems, the outcome's success often relies more on the training data used, rather than the model itself [4]. Hand-collecting and labelling this data, whilst ensuring diversity and accurate annotations for supervised machine learning can be a time-consuming and expensive endeavour [5]. Furthermore, such human-related computer vision tasks introduce additional ethical concerns such as GDPR infringement [6], and require additional attention to bias and model fairness [7]. Fortunately, computer-generated data for machine learning provides a solution to these problems. Thousands of perfectly labeled, diverse, and challenging training data can be automatically generated whilst negating any potential privacy concerns that come with human-related computer vision. Several such datasets have existed outside the field of face-related computer vision for some time [8, 9, 10, 11, 12, 13, 14], but it is only until 2021 when Microsoft/Wood et al. released the first fully synthetic face dataset for face parsing and landmark localization [4].

Wood et al.'s novel dataset consists of 100,000 synthetic face images at 512x512 resolution and consists of diverse faces and lighting with challenging poses and expressions. Their report compares the effectiveness of a segmentation model (UNet [15]) trained on the training portion of popular real-face datasets in the literature against the same model trained on their computer-synthesized data.

Method	Skin	Nose	Brows	Eyes	Upper lip	Inner mouth	Lower lip	Overall
Guo et al. [16]	93.8	94.1	80.4	87.1	75.8	83.7	83.1	90.5
Wood et al. (real)	95.1	94.7	81.5	87.6	81.6	87	88.9	91.6
Wood et al. (synthetic)	95.1	94.5	83.5	87.3	82.3	89.1	89.9	92

Table 1: The results of Wood et al.'s research, tested on the popular Helen dataset.

Wood et al. demonstrated in their paper that a model trained solely on synthetic data could compete with two of the most popular real-face datasets in the literature: Helen [18] and LaPa [19]. However, these results although promising, do not provide a compelling conclusion on how this dataset can perform when tested in the wild. Helen and LaPa, although said to be in the wild by Wood et al., largely consists of portrait, frontal-view, and near-centered images with limited context information [20] (see Table 2), lacking the variability in yaw and subject displacement found in real-life, in-the-wild scenarios; therefore the effectiveness of models trained on this dataset in the wild could still be questioned. Furthermore, their methodology's description lacked important details for reproduction, such as the hyperparameters used in training (number of epochs, batch size, loss function), or the actual training time itself - details of which could not be investigated as the code was also not presented. Moreover, the consistency of the results was not explored across different segmentation models. As such, the interactions and performance of synthetic training data with other popular methods remain unclear.

Benchmark	Images	In the wild	Yaw≥ 30°	Yaw≥ 60°
Helen	2,000	Yes	120	4
LaPa	18, 176	No	3,961	194
iBugMask	21,866	Yes	14,692	6,880

Table 2: A comparison of the variability between discussed facial datasets [17].

1.1 Specification, Aims, and Motivations

My research is important as it aims to contribute to the community by addressing some of the problems present within Wood et al.'s research, providing further insights into the effectiveness of the first fully synthetic dataset for face parsing, and determining if the benefits highlighted in my introduction of using synthetic data for face parsing can finally be reaped.

In-the-wild Representation - My research must provide a more challenging in-the-wild testing environment than the ones used by Wood et al. In Wood et al.'s research, Helen and LaPa were used

as the testing sets. And although they claimed it to be an in-the-wild evaluation of their synthetic data's performance, I do not believe these datasets provide a true representation of an in-the-wild environment, as discussed in this paper's introduction. In my research, I aim to present a conclusive analysis of the performance of segmentation models trained using the novel synthetic training data presented by Wood et al. in the wild, therefore a challenging in-the-wild benchmark: iBugMask [20] (see Table: 2) will be used as the testing set in my research. To compare how the synthetically trained models perform against the real-face alternatives when tested in the wild, the performance of models trained using synthetic data will be evaluated against the same models trained on Helen, LaPa, and a domain-mixed dataset consisting of both synthetic and real-face images on iBugMask. Helen was chosen as a real-face benchmark as it is the most popular and understood benchmark in the field of face-related computer vision, thus allowing my results to be easily compared with similar research in the literature. LaPa was similarly chosen due to its frequent use in other research [4, 17], but also due to it containing considerably more challenging samples and a higher quantity of samples than Helen (see Table: 2), thus potentially creating more well-trained models and better competition for the models trained using Microsoft's dataset.

Label Adaptation - My research must determine the effects label-adapting had on the synthetically trained models' face-parsing predictions. Wood et al.'s paper acknowledged how previous methods narrowed the domain gap between real and synthetic training data by mixing real data with synthetic data [21, 4]. Wood et al., however, describe how they narrowed this domain gap by instead generating ultra-realistic and diverse face models, achieving competitive accuracy without the inclusion of any real data when compared against the same model trained on Helen and LaPa (see Table: 1). Stated in their paper: "we do not perform any of these techniques and instead minimize the domain gap at the source, by generating highly realistic synthetic data". Furthermore, throughout Wood et al.'s paper, it is suggested that the impressive results presented are obtained without using any real data at all. However, this is in fact untrue, as a crucial step in their methodology involves the use of real data: the label adapter [1]. In their research, in order to teach the label adapter how to adapt between synthetic and real images, real-face images were used. Moreover, the magnitude of the effect this label adapter had on the results obtained in the face-parsing experiments presented by Wood et al. are not discussed, and pre-adaptation prediction results from the face-parsing experiments are omitted. Therefore, in my research, I aim to provide the community with results pre-and post-label adaptation, sharing insight on how crucial this stage is for synthetic and real-face data.

Real Data - My research must explore what effects domain-mixing real faces and synthetic faces into the training pool has on in-the-wild face-parsing accuracy. In contrast to the idea of forgoing real-face data conveyed throughout their paper, Wood et al. found success in their results by using real-face images in the training of a label adapter network. Due to this, it begs the question: if success was found by using synthetic data post-training of the initial model (during the training of the label adapter), why not also use it in the initial model training? Although the synthetic training data, due to its controllable diversity, performed well in Wood et al.'s research, I believe that the forgoing of any real face data in the initial model training portion of Wood et al.'s research was a missed opportunity. Due to the success Wood et al. found by using real-face data in the training of their label adapter, and due to the previous success in the literature acknowledged by Wood et al., which used real data alongside synthetic data (that is less realistic than what has been released by Microsoft) [21, 22], I hypothesize Microsoft's new data could not only compete with Helen and LaPa (as presented in Wood et al.'s research and in Table: 1) but actually outperform them when real-face data is also mixed in with the initial models training set. Because of this, I will introduce a new domain-mixed dataset, consisting of real and synthetic images, to explore the performance of mixing real and synthetic data in the initial model training.

Different Models - My research must explore the effectiveness of different popular segmentation models with Wood et al.'s synthetic training data. The consistency of the results was not explored

[1] What a label adapter is, and how one works are described in detail in Chapter: 2.2

across different segmentation models in Wood et al.'s paper. As such, the interactions and performance of synthetic training data with other popular methods remain unclear. In my research, four segmentation models popular in the literature: DeepLabV3+ [23], UNet [15], FPN [24], and MobileNetV2 [25] will be used in this evaluation to explore the interactions that synthetic face data has with different models. UNet, DeepLabV3+, and FPN were chosen due to their success seen in related fields (UNet:[4, 26], DeepLabV3+:[27, 28], FPN:[29]). In order to provide a contemporary conclusion on the state of synthetic face data in machine learning, the inclusion of these high-performing state-of-the-art models in my research is essential. MobileNetV2, however, was chosen to benchmark the performance on a model tuned for mobile CPUs, due to the flourishing domain area of face-related computer vision in applications used in VR [30] and mobile phones [31]. Furthermore, the performance of these models on Helen, LaPa, and Wood et al.'s are either sparse or non-existent in the literature; as such, my results should be beneficial to the community.

Quantity vs Quality - My research must provide a better understanding of the quality of the used training material than what was presented by Wood et al. As can be seen in (Table: 1), Wood et al.'s dataset achieved impressive results when compared to the Helen training set when tested on the Helen test set. To achieve these results, however, Wood et al. used a 100,000-image dataset for training, compared to the 2,000-image dataset used in the Helen training set - making these results a lot less impressive. Furthermore, this also raises the question of the actual quality of the training data presented by Wood et al., as it is possible that they were only able to compete with Helen due to its dramatically increased amount of training data. Because of this, in my experiments, I will use a 2,330 sample of all datasets for training and validation to allow a fair evaluation of the quality of the training data to be made. An exception to this in my experiments will be LaPa, in which the full dataset will be used for training - to see if Wood et al.'s dataset can outperform a real-face dataset of the same size (Helen) and a bigger one (LaPa).

Reproducible Results - My research must provide the community with detailed explanations and open-source code. The experiments conducted in Wood et al.'s methodology's lacked important details for reproduction such as the hyperparameters used in training (number of epochs, batch size, loss function), or the actual training time itself - details of which could not be investigated as the code was also not presented.

The success of this research will be determined by how well each of these points is addressed. My results must convincingly answer if synthetic face data can compete or even outperform Helen and LaPa in the wild; determine the magnitude the label adapter has on the face-parsing performance, determine what effect mixing real data in the initial model training has; determine which segmentation model performs best in the task of in-the-wild face parsing; determine the quality of Microsoft's training data; and have reproducible and fair experiments. To ensure reproducible results, all training and testing methods must be detailed and diagrammed, all parameters and hyperparameters must be available and annotated, and the code must be readable and well-documented. To ensure fair and concrete experimentation and comparisons, models trained using Helen and LaPa must be in line (+/- 15%) with similar methods in the literature before they can be tested on iBugMask.

2 Design

The structure of this research can be divided into two phases: Training (see Figure: 1) and Testing (see Figure: 2). During the training phase, all discussed models will be trained on datasets from different domain spaces. During the testing phase, those same models will be tested in the wild using iBugMask. The training phase can be divided into two further phases: Model Training and Label Adapter Training.

2.1 Model Training

During this phase, four copies of DeepLabV3+, UNet, FPN, and MobileNetV2 will be implemented using Pytorch. Each set of copies will then be trained on Helen, LaPa, synthetic data, and domain-

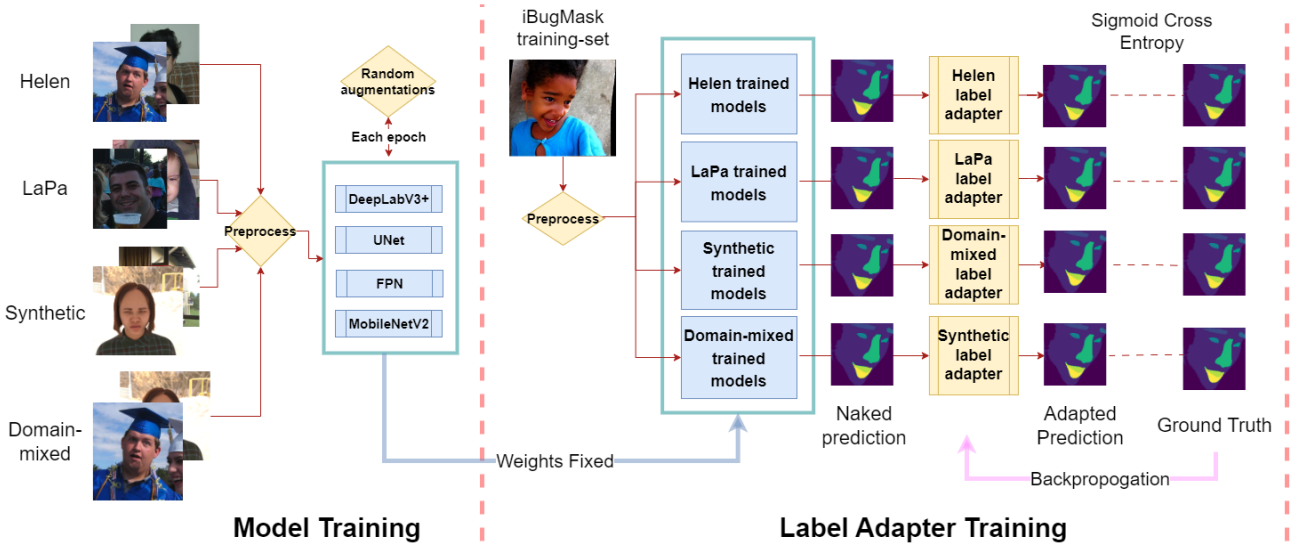


Figure 1: Left: Flow of the initial "Model Training", in which copies of DeepLabV3+, UNet, FPN, and MobileNetV2 are trained on Helen, LaPa, Wood et al.'s dataset, and a new domain-mixed dataset. Right: Flow of the subsequent "Label Adapter Training", in which models from "Model Training" are used to generate naked predictions from iBugMask's training set for use in the training of designated label adapters.

mixed data. Images from these datasets will undergo preprocessing before entering the model, in which they will undergo regularization by face alignment and cropping via the landmark coordinates given in each dataset, as well as normalization to the ImageNet standard. Following Wood et al., due to their successful results, the images will then undergo random augmentations each time they are retrieved from the dataset, making the model see a different set of images every epoch, thus increasing the model's ability to learn as the epochs go on.

2.2 Label Adapter Training

To help bridge the domain gap between the synthetic and the in-the-wild data, I will implement a label adapter network. Label adaptation was a method used by Wood et al. to help adapt predicted labels from the synthetic domain space into the domain space of real face images. Simply, it is a model trained on the predictions of a previous model (trained from images in a subsequent domain space), which has been fed images from a target domain space (see Figure: 1). In my research, the "previous model" will be the Helen, LaPa, synthetic, or domain-mixed trained models, which will be fed images from iBugMask's training set, as images from this lie within the target domain space: real-face and in-the-wild.

Wood et al.'s research described label adapting as helping address the systematic differences between computer-generated labels and hand-generated labels. Their synthetically-trained model's landmark localization results of "common" images experienced a 2.24 decrease in NME, and their real-data-trained model experienced a 0.45 NME decrease. As such, I will implement a label adapter to address differences between the computer-generated and hand-labeled annotations. Additionally, the number of challenging rotations (see Table: 2) present within LaPa and Helen is significantly lower than in iBugMask, and their representation of an in-the-wild environment is far worse. Therefore, I believe the inclusion of a label adapter also has the potential to increase the accuracy of the Helen and LaPa-trained models.

During the Adapter training phase of this research, the models previously trained on Helen, LaPa, synthetic data, and domain-mixed data will be fed images from iBugMask's training set. The predictions made by these models along with the corresponding ground truth will be used in the training of a label adapter. Because the adapter is adapting from the specific subsequent domain space found within Helen, LaPa, synthetic data, or domain-mixed data, this label adapter will be specific to the group of models trained on that dataset (see Figure: 1). To ensure the adapter cannot cheat, only

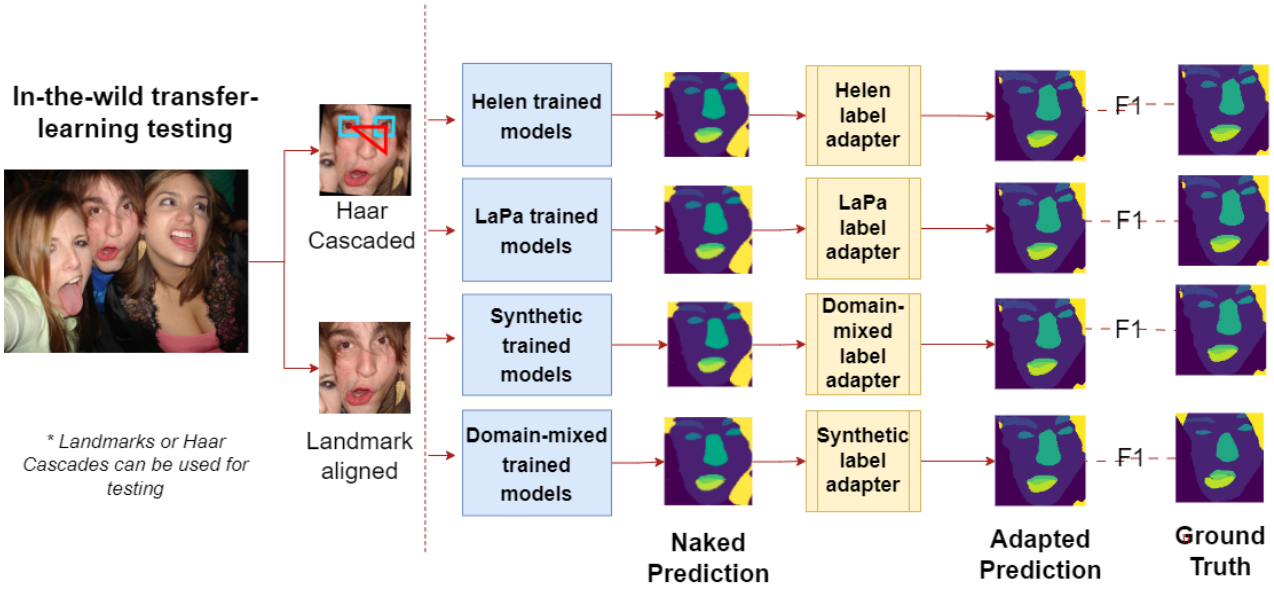


Figure 2: Flow of the "Testing in the Wild" phase, in which images from iBugMask's test set are preprocessed and passed into the models trained in "Model Training" which generates "Naked Predictions" which are then adapted via the label adapters trained in "Label Adapter Training". These adapted outputs are then evaluated against the original image's ground truth via F1.

images from iBugMask's training set will be used.

2.3 Testing in the Wild

During this phase, all models will be tested in the wild via iBugMask's testing set before and after label adaptation. Faces from iBugMask's test set will be aligned using either the given landmark coordinates as done in [4], or intuitively via Haar Cascades and Euclidean geometry, inspired by [3]. The option to intuitively retrieve the coordinates of the facial regions of interest (RoIs) will be included to provide a better simulation of an in-the-wild environment, as the landmark coordinates of the RoIs are not freely given in reality. Results using this method will not be used in my paper, to allow my results to be more comparable to what has already been conducted by Wood et al., but it will be available as an option in the code repo for further research. Once aligned, the faces will be fed into the models trained on Helen, LaPa, synthetic data, and domain-mixed data. The predictions from those models will then be fed into the corresponding label adapter, whose predictions will then be evaluated against the ground truth using F1 score. F1 score is chosen as the evaluation metric due to its frequent use in the literature in related tasks [4, 17, 20]. Additionally, F1 is a preferred evaluation metric in images like faces, where semantic categories such as skin occupy a disproportionately large area of the segmented image. The results from this phase will reveal which models exhibited the highest transfer learning performance in the wild, thus a conclusion can be drawn on the state of the art of synthetic face data for face parsing, and whether it can compete, or outperform and replace real face data.

2.4 Experiment Success Criteria

To ensure concrete and fair testing throughout these experiments the following success criteria are outlined:

1. Models will be implemented as described in their original papers. This ensures that their results are a true representation of the model's performance.

2. To ensure fairness, images from all datasets will be preprocessed in the same fashion: aligned using the given landmarks and normalized to Imagenet standard. The given landmarks will be used for alignment in this research, as it ensures the interior semantic components of the face are similarly represented across all images.

3. All 4 sets of 4 models will be trained on the testing portion of the 4 datasets and tested on the training portion if it exists. The results from these tests must be in-line (+/- 15%) with what is found in the literature. This ensures that I have trained the models with the dataset correctly, thus the results from “Testing in the Wild” will be an accurate representation of the training dataset’s performance.

4. A label adapter network will be trained on the previous model predictions using images from iBug-Mask’s training set. To prevent cheating, only images from the training set will be used. Results post-adaptation must be similar (+/-15%) to pre-adaptation results. This ensures that the label adapter is adapting the predictions domain space correctly, and is not corrupting the original predictions.

5. All models will be tested pre and post-adaptation on images from iBugMask’s test set. Images from iBugMask will have a choice between being aligned with either the given landmark coordinates or using Haar Cascades. Haar Cascades are included to provide future research to conduct tests using methods available in the wild. Pre and post-adaptation results are included as not all models may benefit from adaptation (as described in [4]), as such, this ensures the best results from each model can be evaluated. The inclusion of pre and post-adaptation results also allows an investigation on the magnitude of label adapting on the final results, a topic not discussed by Wood et al.

3 Methods and Implementation

During this section, I will describe the methods and implementations of the “Model Training”, “Label Adapter Training” and “Testing in the Wild” phases of research outlined in Section: 2.

3.1 Model Training

Model training was the first phase of development. During this phase, all datasets were downloaded and all models were implemented and trained. Note that all development was undergone using a single device with an AMD Ryzen 5 with 8 Logical cores, an NVIDIA GTX 1650 Mobile with 4GB of VRAM, and 20GB of physical memory.

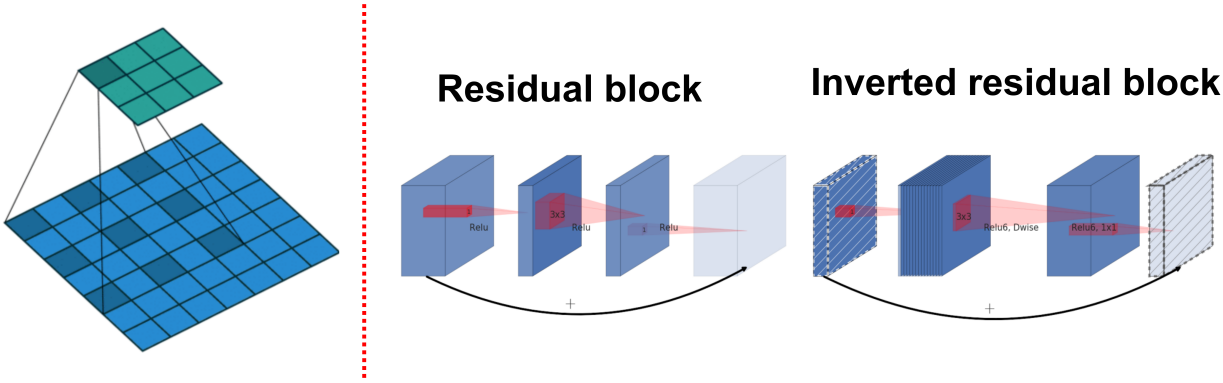


Figure 3: Left: Example of a dilated convolution, as used in DeepLabV3+’s architecture - image taken from [32]. Right: MobileNetV2’s inverted residual structure - image taken from [33]

3.1.1 Models

The most popular segmentation models in the literature all follow the same “encoder-decoder” design, used in the FCN model [34] in 2015. The encoder stage first determines *what* (semantic component) is in the image, and the decoder stage determines *where* that component is within the image. Generally, during the encoder stage, images are fed into a series of convolutional layers, ReLU activation functions, and max pooling layers - progressively shrinking the image down into its semantic components. Then during the decoder stage, the semantic components pass into a series of deconvolutional

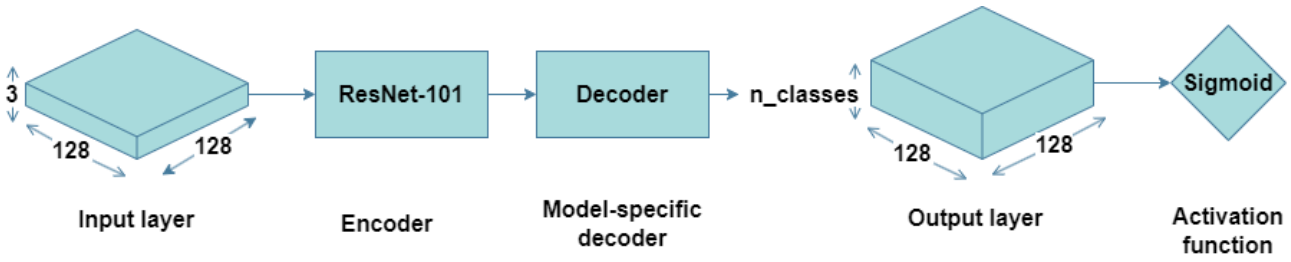


Figure 4: Architecture of models used in “Model Training”

layers, where upsampling is performed to determine where that semantic component is located.

The models used in my research all build upon this design and address certain issues present within the FCN model:

1. **UNet** - Introduced by Ronneberger et al. [15], UNet builds upon the FCN by allowing fewer training features to be given [35] and introduces a symmetrical encoder-decoder cone with skip connections between opposing sides of the cone - recovering fine-grained details in the resulting prediction, thus increasing the prediction resolution and tackling the problem of vanishing gradient [36].

2. **DeeplabV3+** - Introduced by Chen et al. [23] in 2015, introduces dilated convolutions within the encoder-decoder architecture - increasing the kernel’s field of view, at the same computational cost (See Figure: 3).

3. **FPN** - Introduced by Lin et al. [24] in 2016, the Feature Pyramid Network for object detection (FPN), introduces “Feature Pyramids”. A component often used in recognition systems for detecting objects at varying scales. I have specifically included this model as there exists a large disparity between the scale and representation of semantic components within a face image. For example, the eyebrows and inner mouth area occupy a significantly smaller surface area than the facial-skin region. As such, I hypothesize that due to the inclusion of Feature Pyramids, this model will perform the best.

4. **MobileNetV2** - Introduced by Sandler et al. [33] in 2018, the MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual modules (See Figure: 3), and uses lightweight depthwise convolutions to filter features in the expansion layer. These features, more specifically the inverted residual bottleneck layers, allow a particular memory-efficient implementation, which is important in mobile applications and helps them run on devices of varying levels of computational power. In my research, I use the encoder outlined in the MobileNetV2 paper, whilst using the decoder outlined in [15].[2]

Due to the vast amount of training required in this research, in order to easily allow fleets of segmentation models to be quickly implemented and trained, all whilst ensuring the models used are implemented as described in their papers, the Pytorch API: “Segmentation Models Pytorch” [37] was used. This API provides access to pre-trained segmentation models implemented as described in their relevant papers. Using this API, I can then set the same encoder and initial encoder weights to be used across all models, increasing the equality and fairness in the training of the models.

Regarding the structure of the models used (see Figure: **??**), Wood et al. used a ResNet-18 (18-layer deep Resnet) encoder for their model. Resnet, introduced by He et al. [38], is an encoder that

[2]MobileNetV2 and FPN were swapped from LRASPP and FCN originally planned in the Literature Review. This was because I believe FPN to be an especially suitable model for this task, and MobileNetV2 due to its inclusion in the “Segmentation Models Pytorch Library”.

addresses the problem of vanishing gradient that occurs within deep neural networks, via skip connection addition operators between convolutional layers. I believe, however, that a deeper encoder will be able to learn facial components at a deeper level of abstraction, thus increasing prediction accuracy and potentially generating a more fine-grained output. Liu et al., the contributors of the LaPa dataset, found success in their research [39] via the adoption of a ResNet-101 encoder, which managed to achieve an overall F1 score of 92.4 on the Helen test set, compared with Wood et al.'s 92.0. Because of this, a Resnet-101 (101-layer deep Resnet) encoder was chosen for all models in my research except MobileNetV2, which used the specific "MobileNetV2" encoder included within the Segmentation Models Pytorch library. To allow a faster convergence during training time [40], all models were pre-trained using ImageNet [41] - a popular large-scale database often used in model pre-training for visual object recognition tasks. All models had a 3×128×128 wide input layer. Input images were not normalized to greyscale, and instead, a 3-channel input layer was used to represent the RGB layers present within the input image as this is what is expected for ImageNet-trained models. 128×128 was chosen as the resolution because at this resolution, the input is large enough to allow the model to capture all semantic components entirely but not too large as to dramatically increase the training time. The output layer was n_classes×128×128 wide, which allowed probabilities for each semantic class to be generated, thus creating a more expressive output. Because of this, as previously done by [4], all ground-truth images were one-hot-encoded. Finally, As this problem is a multi-label classification task, Sigmoid Cross-Entropy was chosen as the criterion. Sigmoid, unlike losses intended for multi-class problems such as Softmax, calculates the probability for each vector component (semantic category) independently, and therefore independent for each output layer. Simply, this means elements belonging to a certain class do not influence the predictions of another class. As such, the final layer of each model included a Sigmoid activation function.

3.1.2 Dataset Processing

To help address the differences in how the different datasets used are organized, each dataset was used and processed slightly differently to achieve a fairer level of training.

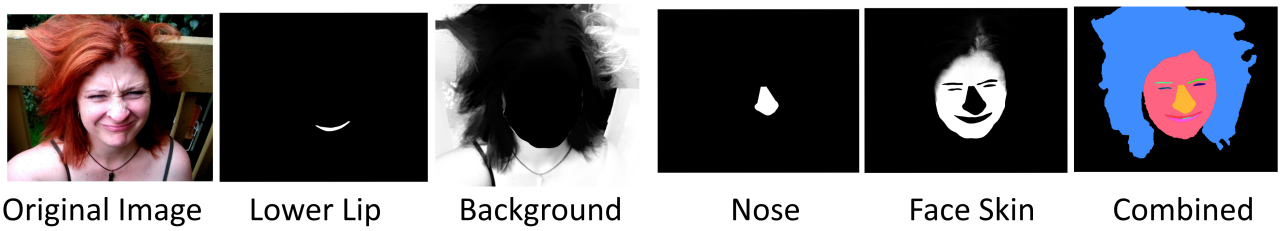


Figure 5: Individual masks for each semantic component presented in Helen. The "Combined" image demonstrates the combined layers of the label after each pixel has been assigned a class value. Note that the final combined image is greyscale, with pixels ranging from 0 - 11; for demonstration purposes, it has been coloured in post-processing.

Helen - Unlike LaPa, iBugMask, and Wood et al.'s dataset, Helen was not originally intended for face parsing. Because of this, a modified version created by Smith et al. [42] was used in this research due to its inclusion of segmentation masks for face parsing. However, unlike LaPa's, iBugMask's, and Wood et al.'s segmentation masks, the masks present within Helen are not encoded within a single greyscale image with each pixel denoting the class value:

0: background, 1: face skin, 2: left brow (viewer side), 3: right brow, 4: left eye, 5: right eye, 6: nose, 7: upper lip, 8: inner mouth, 9: lower lip, 10: hair.

Instead, each image within Helen corresponds to 11 individual segmented images - for the 11 semantic classes within the original face image (see Figure: 5). In each segment, each pixel denotes the probability (between 0 and 255) that the pixel belongs to that semantic class. For faster normalization and augmentation, however, these different segments were merged into a single greyscale

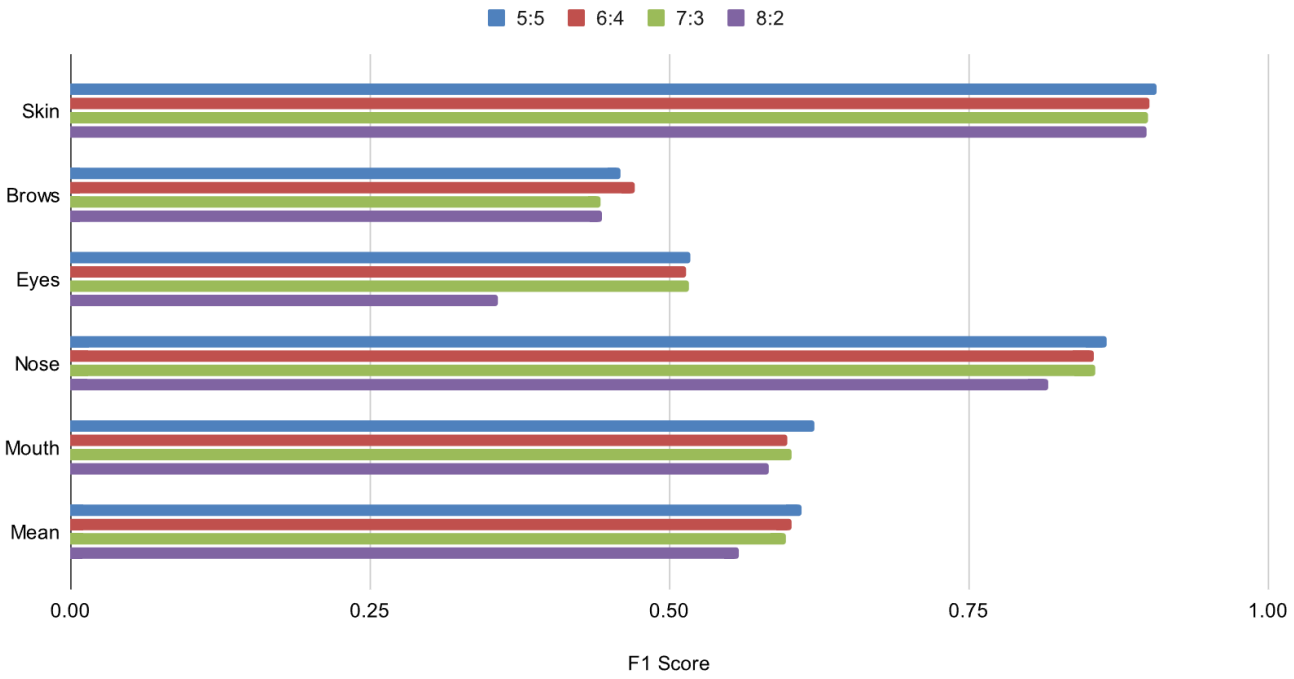


Figure 6: An improvement of F1 in datasets with higher quantities of real data mixed in. The 5:5 split exhibited the highest F1 across important face classes when tested on iBugMask's test set (pre-adaptation results).

ground-truth image. During merging, each layer goes through activation, which changes each mask's pixels above 127 (above 0.5 certainty that that pixel is correctly labelled), into the semantic label it corresponded with (0 to 10). After all processes are done on the ground-truth image, it is then one-hot-encoded to match the n_classes-wide output layer of the model. Finally, following the training, validation, and testing split used in [42], I use 2,000 images for training, 230 images for validation, and 100 images for testing Helen.

Synthetic - The synthetic data provided by Microsoft, unlike Helen, LaPa, and iBugMask, featured 20 different semantic classes. This means the preprocessing pipeline for images taken from Wood et al.'s dataset involved the removal of the extra 9 classes for a fairer comparison. The extra 9 classes were changed to a new 12th class: 11: ignore. 11 was then used as the ignore index, and had a weight of 0 when passed into the loss function. Wood et al.'s synthetic dataset in its entirety is 32 GB and consists of 100,000 images. To provide a fair evaluation of the quality of this training data in comparison to the other datasets, a 2,330 image sample was taken from the full dataset, and the same training, validation, and testing split as Helen was used.

Domain-Mixed - Determining the ratio of real to synthetic data to be included in the domain-mixed dataset involved 4 trials, in which 4 separate DeepLabv3+ models were trained over 30 epochs, and then tested on the iBugMask test set. Each model was trained using 2,000 training images and 230 images for validation and consisted of synthetic images taken from Wood et al.'s and real-face in-the-wild images sampled from LaPa. Although images from iBugMask's training set would most likely provide better training material for this task as opposed to images taken from LaPa, I have chosen to take images from LaPa to provide a fairer comparison between the other datasets. Choosing to mix synthetic images from LaPa in the creation of this domain-mixed dataset will allow us to see the negative or positive effects the act of domain-mixing images from the pure datasets had in comparison to the unmixed versions. The ratio of synthetic to real images for each model was 8:2, 7:3, 6:4, and 5:5. These models were then tested on iBugMask's test set. The results of this experiment revealed a consistent improvement of F1 score manifesting across splits with a higher proportion of real images, leading to the 5:5 dataset being chosen to represent the domain mixed dataset. The results of this experiment can be seen in (Figure: 6). This outcome is most likely because iBugMask solely consists of real-face images, thus the inclusion of more real-face images into the training pool shortens the

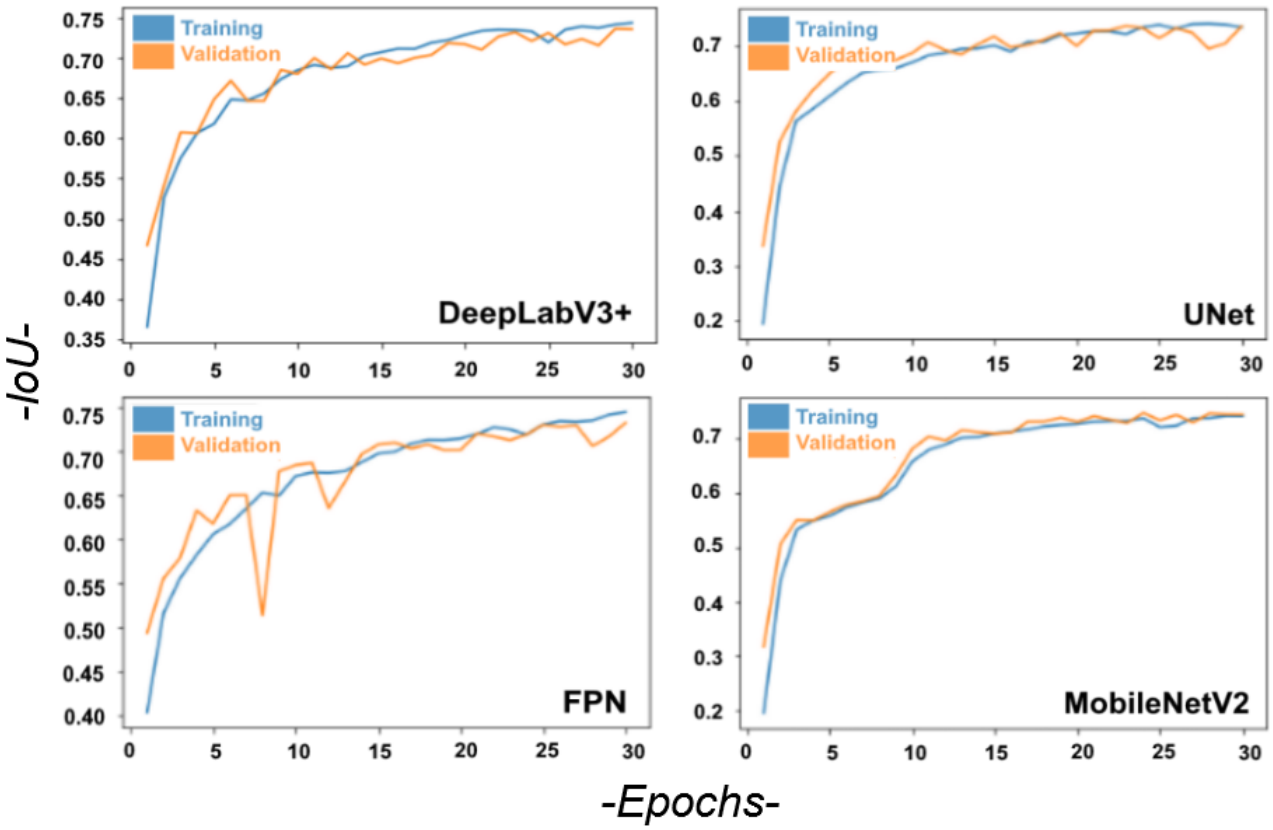


Figure 7: Plotting of IoU against Epoch count during the training of models on the Helen dataset. Optimum convergence begins around epoch 7. Due to the augmentations applied each epoch, overfitting does not occur - evident by the minimum separation between the training and validation lines.

domain gap the model must overcome when transferring to iBugMask's test set. Finally, the domain-mixed dataset also followed the same training, validation, and testing split used in Helen, to allow for a fair comparison.

LaPa - Due to LaPa's entire size being only 2.3 GB was used in its entirety. The split used in training models using LaPa followed the one presented in [19], consisting of 18,000 images for training, 2,000 for validation, and 2,000 for testing. Because of this, models trained on LaPa should act as a challenging comparison to the other models, due to their increased amount of training material. Furthermore, Wood et al. argued in their paper that synthetically generated data allows for more expressive and diverse face images, meaning that each training image could potentially be more valuable than the training images taken from a real-face dataset like Helen or Lapa. As such, it should be interesting to see how the synthetic data competes against a real-face dataset of its own size (Helen) and larger size (LaPa).

All Images - Due to the limited disk space on the device used, all processing was done during training time. The images taken from all datasets underwent the following processing pipeline:

1. **Alignment** - To allow easier learning for the models, all facial regions of interest (RoIs) are aligned to approximately occupy the same space in every image. The original image is cropped around the RoIs, using the landmark coordinates given in every dataset.

2. **Reshaping** - All images are reshaped to 128x128. Reshaping is used instead of cropping, as cropping would remove critical semantic components within the image.

3. **Augmentations** - Performing augmentations at training time is especially beneficial. It ensures that every time the image is pulled from the dataset, a new random set of augmentations is

applied. This means that during each epoch, the model is seeing an entirely new set of images - increasing the amount it can learn per epoch. Albumentations [43] was used as the augmentation library, as it easily allows you to apply the same augmentations to the ground truth and original image. The augmentation pipeline each image and ground truth mask goes through consists of a rotation between (-18°, 18°) with a probability of 1, a perspective shift with a probability of 0.5, addition of Gaussian noise with a probability of 0.2, one of: (contrast limited AHE, brightness or gamma augmentations) with a probability of 0.9, one of: (sharpen or blur augmentations) with a probability of 0.9, and one of: (contrast or hue shifts) with a probability of 0.9.

4. **Normalisation** - As all models were pre-trained on Imagenet, for optimal results, all images were normalized to Imagenet standard (mean = [0.485, 0.456, 0.406] and stanard_form = [0.229, 0.224, 0.225]) before entering the model.

5. **One-hot-encoding** - To match the num_classes-wide output layer, and allow for probabilities for each semantic class to be compared in the criterion, all ground-truth labels are one-hot-encoded.

3.1.3 Training

Training all 16 models (4 models per 4 training datasets) took a total time of 12 hours 36 minutes, with an average training time of 46 minutes per model. Lapa, being a much larger sized dataset averaged a training time of 1 hour per model. Due to the large number of models used in this research, the hyperparameters used for model training were chosen to balance both accuracy and training time. As such, training was limited to 30 epochs, as during this time all models reached convergence to an acceptable degree - with optimum convergence beginning approximately at epoch 7 across all models (see Figure: 7). A batch size of 24 was used, as this allowed for faster training time and a thorough navigation of the optimization landscape. Furthermore, any batch size higher than 24 resulted in memory issues. NAdam [44] was chosen as the optimizer, with an initial learning rate of 0.001. NAdam is an improvement upon the Adam optimizer, by including Nesterov's accelerated gradient [45] (a form of "improved momentum" [44]). In [44] Nadam was able to reach a more optimal solution faster than the optimizer used by Wood et al. (Adam), as such Nadam was chosen as the optimizer for all models in my research. Following the successful processes conducted by Liu et al. [39], who contributed the LaPa dataset, weighted sigmoid cross entropy loss was used as the criterion, with weights determined from the proportion of each class in the dataset. Classes with a higher representation were weighted proportionally lower than those with a lower representation. Weighted Sigmoid-Cross-Entropy was implemented to further help tackle the large disparity between the scale and representation of semantic components within a face image. A graphing of IoU per epoch can be seen in Figure: 7.

3.1.4 Results

Following training, models trained on the Helen training set were tested on the Helen test set, and models trained on LaPa's training set were tested on LaPa's test set. This was done to make sure the training images from the datasets were handled correctly, and that the results from training are in line with similar results in the relevant literature before moving on to the In-the-Wild Testing with iBugMask.

Model	Skin	Hair	L-Eye	R-Eye	U-lip	I-mouth	L-lip	L-Brow	R-Brow	Nose	Mean
DeepLabV3+	0.960	0.730	0.680	0.698	0.771	0.692	0.804	0.679	0.713	0.922	0.765
FPN	0.963	0.761	0.654	0.210	0.784	0.715	0.825	0.696	0.724	0.897	0.723
Unet	0.967	0.759	0.710	0.717	0.795	0.727	0.836	0.705	0.729	0.933	0.788
MobileNet2	0.967	0.755	0.710	0.716	0.795	0.727	0.829	0.702	0.730	0.896	0.783
Average	0.964	0.751	0.688	0.585	0.786	0.715	0.823	0.695	0.724	0.912	0.765

Table 4: Results from the LaPa test set from models trained on the LaPa training set. No relevant comparisons can be made from results in the literature, as all papers using LaPa involve novel technologies not used in this research.

Model	Eyes	Eyebrows	Nose	Inner Mouth	Upper Lip	Lower Lip	Mouth (all)	Face Skin	Average
DeepLabV3+	0.733	0.679	0.922	0.706	0.692	0.776	0.724	0.944	0.779
FPN	0.356	0.711	0.897	0.676	0.701	0.795	0.724	0.943	0.726
Unet	0.747	0.708	0.897	0.662	0.714	0.797	0.724	0.945	0.781
MobileNet2	0.750	0.709	0.896	0.673	0.720	0.798	0.730	0.944	0.785
Average	0.646	0.702	0.903	0.679	0.707	0.792	0.726	0.946	0.768
Zhu et al [46]	0.533	n/a	n/a	0.425	0.472	0.455	0.687	n/a	n/a
Saragih et al [47]	0.679	0.598	0.890	0.600	0.579	0.579	0.769	n/a	0.733
Liu et al [48]	0.770	0.640	0.843	0.601	0.650	0.618	0.742	0.886	0.738
Gu et al [49]	0.743	0.681	0.889	0.545	0.568	0.599	0.789	n/a	0.746

Table 3: Results from the Helen test set from models trained on the Helen training set. Model results are in-line or better than previous results in the literature. Following previous works using Helen, hair, background, and other fine-grained categories are omitted, and the left/right eyebrow and eye, inner mouth, upper lip, and lower lip scores, are averaged to obtain single “Eyes”, “Eyebrows”, and “Mouth (all)” categories.

The models trained on Helen exhibit excellent accuracy, surpassing previous results in related studies. LaPa’s trained models exhibit similar results to Helen, averaging a difference of 0.2 between the Helen and LaPa-trained models. The lower performance displayed in LaPa’s models is expected, due to LaPa’s images having more challenging variations such as yaw (see Table: 2), thus increasing the learning difficulty. The greater challenge provided by LaPa however will most likely mean it outperforms Helen when tested in the wild with iBugMask. Unfortunately, no relevant results from the literature could be found for comparison on LaPa. Microsoft’s research using LaPa, although similar, did not include the Epoch count or training time, therefore I do not believe a relevant comparison against my models trained over 30 epochs on a single device can be made. [50] and similar research all included novel techniques such as visual-linguistic processing, which is far more advanced and exorbitant than what is required and implemented in this research. As such, results from the discussed research are omitted for comparison.

3.2 Label Adapter Training

During this phase, the models previously trained will each be fed images from the iBugMask training set. The segmentation predictions these models produced, along with the ground truth of the original image used, will then be used in training a label adapter for that dataset; teaching the label adapter how to adapt given predictions into an in-the-wild environment.

3.2.1 iBugMask

As iBugMask’s training exists within the same in-the-wild domain space as its test set, iBugMask’s training set was used as the training material to teach the label adapters how to adapt images from a subsequent domain space into a real-face in-the-wild one. iBugMask’s training set also contains considerably more challenging and varied poses than what’s present in Helen and LaPa (see Table: 2), and in-the-wild, group scenarios with multiple subjects such as “party” and “conference” [20], making it incredibly suitable training data for bridging the domain gap into an in-the-wild domain space.

3.2.2 Models

Wood et al. utilized the same network design for both their original model and label adapter: UNet-Resnet-18. Following their research, I also used the UNet model previously implemented in “Model Training” for the label adapter in my research. As such, the model had the same 101-layer deep, ResNet encoder, 3-wide input layer, num_classes wide output layer, Sigmoid final layer, and ImageNet initial weights as previously used in “Model Training”. For use in further research, I have also included an FCN with VGG encoder developed using Pytorch for use as a label adapter in the code repo. This FCN was developed as there is no existing research focusing on the effects and performance of different segmentation models in the task of label adapting, therefore further investigation on this topic would be very interesting. It is, however, not particularly relevant to the focus of this paper, so only the results from a UNet label adapter will be discussed due to UNet’s inclusion of skip connections and therefore higher resolution prediction output.

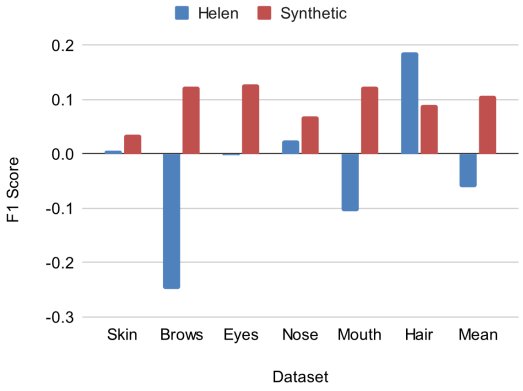
3.2.3 Training

Images taken from the iBugMask training set were fed into the Helen, LaPa, Domain-mixed, and synthetically trained models. Predictions from these models, along with the corresponding ground truth, were then used in the training of a UNet label adapter designated that set of models. The same training parameters as used in "Model training" were used, and the total training time for all 4 adapters was 3 hours.

3.2.4 Results

The effects of label adaptation can be seen in Figure: 8, in which a preliminary test of models trained on Helen and synthetically trained models were tested on iBugMask's test set. From these results, it can be seen that the label adaptation of the synthetic-trained model significantly improved accuracy and tackled dramatic semantic misclassifications of classes with large areas such as facial skin and small areas such as the brows, a phenomenon highlighted in Figure: 13. Wood et al. described a similar recovery of semantic labels in areas such as the chin in their adaptation results in their paper. My preliminary test results showed that models trained on purely synthetic data had an average F1 increase of 0.11 across all facial categories, and improved the accuracy of the predictions of every semantic class post-adaptation. Models trained on Helen, however, did not benefit from label adaptation. Models trained on Helen displayed an average F1 decrease of 0.06. This decrease in accuracy in post-adaptation-Helen results, although unexpected, is most likely due to the level of challenge between iBugMask and Helen being too great for the label adapter to learn how to bridge the domain gap. This suggests that the domain gap between Helen and iBugMask, although both datasets consist of real-face images, could be greater than the one present between Wood et al.'s dataset and iBugMask. This lends great credit to the realism and challenge of the diversity of images achieved by Wood et al.'s dataset, which has allowed this domain gap to be shortened. Full results of every model after label adaption is presented and discussed in Chapter: 6. Finally, as outlined in 2.4, results post-adaptation are well within 15% of pre-adaptation results, thus the next phase of experimentation can begin.

Figure 8: Plotting of the average difference in F1 scores of key facial categories before and after label adaptation from DeepLabV3+ on images from iBugMask's test set.



3.3 Testing in the Wild

After all models and adapters were trained, a transfer-learning assessment was conducted using the full iBugMask test set. Images from iBugMask's test set were fed into the 4 sets of models (see Figure: 2) to produce naked predictions, which were then fed into their corresponding label adapter. This label adapter produced adapted predictions into the real-face in-the-wild domain space, which were then compared against ground truth using F1 score as the metric.

3.3.1 Alignment

In my code, faces from iBugMask's test set can be aligned using either the given landmark coordinates or intuitively via Haar Cascades and Euclidean geometry, depending on the parameters chosen in the testing dataset's creation. Haar cascades are a method of object detection within images that can perform regardless of scale or rotation. As such, they provide the perfect means of cropping a facial region from an image as part of preprocessing. Two of OpenCV's pre-trained Haar Cascade classifiers were used during preprocessing: A frontal face classifier and a profile face classifier. These two classifiers are repeated in the preprocessing pipeline at differing resolutions, allowing for faces

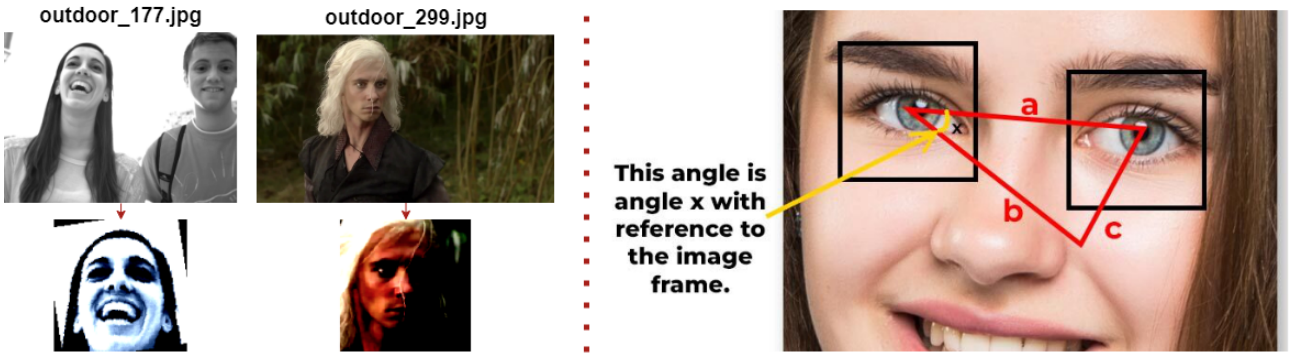


Figure 9: Left: a demonstration of iBugMask test images before and after preprocessing. Right: a demonstration of how the angle of rotation is obtained from a face image taken from [51]

at varying scales and orientations to be more easily detected. Following this, all cropped faces are rotated to ensure all facial regions of interest are in approximately the same space within the model. The angle of rotation x is calculated from a triangle between the eyes of the subject, aligned via the reference frame. Applying the Euclidean distance, we can obtain the value of the angle x (see Figure: 9, [51]).

$$cos(x) = \frac{b2 + c2 - a2}{2bc}$$

4 In-the-wild Results and Evaluation

In this section, I will present and discuss the results of the previously trained models in-the-wild using iBugMask. As per the rest of this paper, F1 score was used as the metric to evaluate the performance of the datasets when passed through the testing pipeline. F1 between the ground truth and predictions is calculated both before and after label adaptation in order to explore the interactions the different training material has with the label adapter. Post-adaptation results appear below pre-adaptation results in a lighter colour in the results table. The best model from these experiments (MobilenetV2-based Unet) was trained on a random sampling of the iBugMask training set (using the same training, testing, and validation split as the other datasets except LaPa), to act as the upper bound in this evaluation. Results for the upper bound can be found at the bottom of Table: 6 and table: 7. Full results can be found in Table: 7, and the best average from each set of models can be found in Table: 6 and Figure: 10, and the best-performing model architecture can be found in Table: 8. Using these results, I will then go back to the aims outlined in Chapter: 1.1 and discuss what has been achieved and what conclusions can be drawn.

Firstly, as a baseline/control to verify the validity of my trained models before continuing, and to allow a comparison between the work conducted in my research and in Wood et al.'s, a separate experiment was done in which the synthetic, domain-mixed, and Helen-trained models were tested on the Helen test set (See Table: 5). As done by Wood et al.'s paper, a separate label adapter was trained using images from the Helen training set, as this dataset lies within the target domain space.

Trained on	Skin	Brows	Eyes	Nose	Upper Lip	Inner Mouth	Lower Lip	Mean
Synthetic	0.930	0.589	0.706	0.903	0.655	0.703	0.773	0.751
Domain-Mixed	0.939	0.693	0.742	0.924	0.692	0.621	0.798	0.773
Helen	0.944	0.679	0.733	0.922	0.692	0.706	0.776	0.779
Wood et al. (Real)	0.951	0.815	0.876	0.947	0.816	0.870	0.889	0.881
Wood et al. (Synthetic)	0.951	0.835	0.873	0.945	0.823	0.891	0.899	0.888

Table 5: Results from the control experiment, in which the synthetic, domain-mixed, and Helen-trained models were tested on the Helen test set.

From Table 7, Helen results from both mine and Wood et al. (2,000 sample in both) have a percentage difference of 12%, which is under the 15% error bound described in my success criteria. Synthetic results from mine and Wood et al. have a larger percentage difference of 17% which is expected as I only used a 2,000-image sample of their dataset compared to their 100,000-image sample. As explained in Chapter: 1.1, this was done to evaluate the actual quality of the training data.

4.1 In-the-Wild Results

Dataset	Skin	L-Brow	R-Brow	L-Eye	R-Eye	Nose	U-Lip	I-Mouth	L-Lip	Hair	Mean
Helen	0.907	0.440	0.464	0.498	0.410	0.855	0.598	0.542	0.655	0.475	0.585
LaPa	0.912	0.444	0.470	0.504	0.415	0.855	0.613	0.554	0.665	0.554	0.598
Synthetic	0.896	0.449	0.435	0.470	0.418	0.862	0.582	0.652	0.676	0.537	0.598
Mixed(50:50)	0.896	0.462	0.464	0.533	0.524	0.876	0.440	0.665	0.697	0.556	0.611
iBugMask	0.912	0.531	0.531	0.499	0.534	0.833	0.607	0.503	0.633	0.522	0.610

Table 6: Best “Average” results taken from each dataset’s assessment. Synthetic-based datasets outperform or compete with both real-face datasets.

Trained on	Model	Skin	L-Brow	R-Brow	L-Eye	R-Eye	Nose	U-Lip	I-Mouth	L-Lip	Hair	Mean	Net Adaption
	DeepLabV3+	0.895	0.421	0.436	0.479	0.482	0.850	0.555	0.510	0.626	0.393	0.565	
	Label Adapted	0.901	0.114	0.245	0.493	0.460	0.875	0.592	0.206	0.576	0.581	0.504	-0.060
	FPN	0.912	0.452	0.476	0.486	0.117	0.856	0.614	0.558	0.670	0.548	0.569	
Helen	Label Adapted	0.903	0.131	0.277	0.488	0.188	0.876	0.626	0.215	0.578	0.546	0.483	-0.086
	Unet	0.906	0.442	0.471	0.510	0.521	0.855	0.606	0.549	0.651	0.398	0.591	
	Label Adapted	0.904	0.126	0.234	0.480	0.450	0.875	0.627	0.201	0.571	0.598	0.507	-0.084
	MobileNet2	0.914	0.447	0.474	0.518	0.519	0.858	0.619	0.552	0.675	0.561	0.614	
	Label Adapted	0.904	0.122	0.237	0.468	0.488	0.883	0.629	0.204	0.578	0.554	0.507	-0.107
	Average	0.907	0.440	0.464	0.498	0.410	0.855	0.598	0.542	0.655	0.475	0.585	
	Average (Label Adapted)	0.903	0.123	0.248	0.482	0.396	0.877	0.618	0.207	0.576	0.570	0.500	-0.084
	DeepLabV3+	0.908	0.427	0.460	0.500	0.509	0.847	0.606	0.540	0.643	0.548	0.599	
	Label Adapted	0.918	0.250	0.353	NaN	0.562	0.875	0.624	0.666	0.702	0.666	0.561	0.025
	FPN	0.912	0.452	0.476	0.486	0.117	0.856	0.614	0.558	0.670	0.548	0.569	
LaPa	Label Adapted	0.917	0.245	0.405	NaN	0.322	0.877	0.623	0.673	0.725	0.674	0.546	0.038
	Unet	0.913	0.450	0.470	0.511	0.515	0.859	0.613	0.565	0.673	0.558	0.613	
	Label Adapted	0.920	0.231	0.364	NaN	0.567	0.877	0.622	0.678	0.731	0.676	0.567	0.017
	MobileNet2	0.914	0.447	0.474	0.518	0.519	0.858	0.619	0.552	0.675	0.561	0.614	
	Label Adapted	0.920	0.227	0.389	NaN	0.565	0.878	0.627	0.670	0.723	0.678	0.568	0.017
	Average	0.912	0.444	0.470	0.504	0.415	0.855	0.613	0.554	0.665	0.554	0.598	
	Average (Label Adapted)	0.919	0.238	0.378	NaN	0.504	0.877	0.624	0.672	0.720	0.673	0.560	0.024
	DeepLabV3+	0.865	0.351	0.364	0.370	0.366	0.793	0.457	0.506	0.595	0.445	0.511	
	Label Adapted	0.901	0.475	0.488	0.522	0.470	0.863	0.592	0.659	0.679	0.536	0.618	0.107
	FPN	0.868	0.229	0.266	0.440	0.436	0.814	0.469	0.529	0.620	0.438	0.511	
Synthetic	Label Adapted	0.899	0.452	0.493	0.523	0.500	0.859	0.594	0.649	0.662	0.529	0.616	0.105
	UNET	0.875	0.117	0.100	0.410	0.376	0.823	0.386	0.103	0.604	0.429	0.422	
	Label Adapted	0.893	0.384	0.258	0.533	0.486	0.859	0.558	0.627	0.650	0.540	0.579	0.157
	MobileNet2	0.866	0.431	0.463	0.100	0.100	0.804	0.500	0.516	0.647	0.461	0.489	
	Label Adapted	0.892	0.484	0.500	0.302	0.216	0.866	0.584	0.671	0.714	0.544	0.577	0.089
	Average	0.868	0.282	0.298	0.330	0.319	0.808	0.453	0.413	0.616	0.443	0.483	
	Average (Label Adapted)	0.896	0.449	0.435	0.470	0.418	0.862	0.582	0.652	0.676	0.537	0.598	0.114
	DeepLabV3+	0.897	0.436	0.449	0.494	0.498	0.837	0.587	0.535	0.648	0.506	0.589	
	Label Adapted	0.895	0.464	0.455	0.534	0.532	0.875	0.413	0.665	0.696	0.539	0.607	0.018
	FPN	0.900	0.433	0.465	0.496	0.496	0.856	0.588	0.541	0.661	0.514	0.595	
Mixed(50:50)	Label Adapted	0.894	0.475	0.463	0.533	0.520	0.874	0.437	0.654	0.687	0.565	0.610	0.015
	UNET	0.896	0.430	0.454	0.496	0.507	0.841	0.585	0.552	0.661	0.512	0.593	
	Label Adapted	0.896	0.438	0.457	0.536	0.528	0.878	0.474	0.667	0.700	0.564	0.614	0.020
	MobileNet2	0.906	0.449	0.477	0.487	0.511	0.852	0.587	0.532	0.669	0.510	0.598	
	Label Adapted	0.899	0.473	0.478	0.529	0.517	0.877	0.436	0.672	0.707	0.556	0.614	0.017
	Average	0.900	0.437	0.461	0.493	0.503	0.847	0.587	0.540	0.660	0.510	0.594	
	Average (Label Adapted)	0.896	0.462	0.464	0.533	0.524	0.876	0.440	0.665	0.697	0.556	0.611	0.018
iBugMask	MobileNet2	0.912	0.531	0.531	0.499	0.534	0.833	0.607	0.503	0.633	0.522	0.610	

Table 7: Results of each model before and after label adaptation tested on iBugMask. The difference in F1 between post and pre-adaptation is calculated and highlighted in the results table under “Net adaptation”. Averaged results from all models pre and post-adaptation for each dataset can be found in “Average” and “Average (Label Adapted)” respectively. Upper bound results can be found from the “iBugMask” row, in which the highest performing model (MobileNetV2-based UNet) was trained on the training portion of iBugMask. Any NaN values were treated as 0 during averaging.

Table: 6 and Figure: 10 use the best average results from each dataset’s models. “Helen” and “LaPa” both use the pre-adaption results, whereas “Synthetic” and “Mixed(50:50)” both use the post-adaptation results. From this, we can see that the hypothesis regarding the domain-mixed dataset laid out in Chapter 1.1 is correct. The domain mixed dataset provided the best training material for

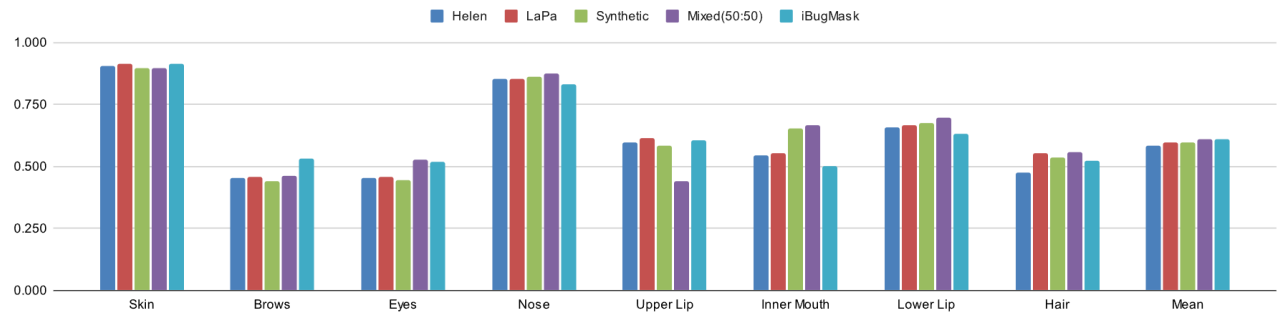


Figure 10: Plotting of the best average results taken from each dataset's assessment. Eyes and brows have again been averaged into single categories for easier digestion.

in-the-wild face parsing with an average of 0.611, beating even the upper bound, which followed in second place with 0.610. Following in third place was LaPa with 0.598, followed by Microsoft's dataset with 0.598 in fourth place, and lastly by Helen with 0.585 in last place.

In the wild, the synthetically trained models struggled with certain semantic categories and excelled in others. From Figure: 10, we can see that the synthetic and domain-mixed models performed the highest in "Nose", "InnerMouth", and "LowerLip", even beating the upper bound. However, in the "UpperLip" category, both the synthetic and domain-mixed models struggled greatly, performing far worse than the real-face alternatives. This is most likely due to a disparity between the representation of this semantic category between Wood et al.'s and iBugMask's images.

The disparity present between the real-face and synthetic data performances can most likely be attributed to the diversity and challenge achieved in Wood et al.'s dataset. From these results, we can see a progressive improvement in mean F1 as more challenging and diverse training material is used. Helen, as seen in Table: 2, provided the least amount of variation in their images, as such it placed 4th. LaPa and Wood et al.'s dataset, as seen in Table: 2, introduces more challenging and varying poses, placing 3rd and 4th with almost identical F1 scores. Finally, the combination of the two most challenging datasets (LaPa and Wood et al.'s), as seen in Table: 2, resulted in the shortest domain gap between it and the test set - resulting in 1st place being given to the domain-mixed dataset.

Returning to the first aim outlined in Chapter 1.1 of testing Wood et al.'s dataset in a truly in-the-wild environment, I believe these results provide the community with a detailed evaluation of the in-the-wild performance of each set of training data and have shown that mixing synthetic faces with real faces via the domain-mixed dataset provided the best training material. I believe this is due to the domain-mixed dataset shortening the domain gap between the subsequent datasets and iBugMask's in-the-wild representation by the greatest amount. Not only does the domain-mixed dataset's images have subjects taken from the fully synthetic dataset, which features a high variety of expressions, poses, and environments similar to iBugMask's, but also includes real-face images, helping models trained on it to shorten the domain gap between it and iBugMask's test set greater than all other datasets (see Figure: 11).

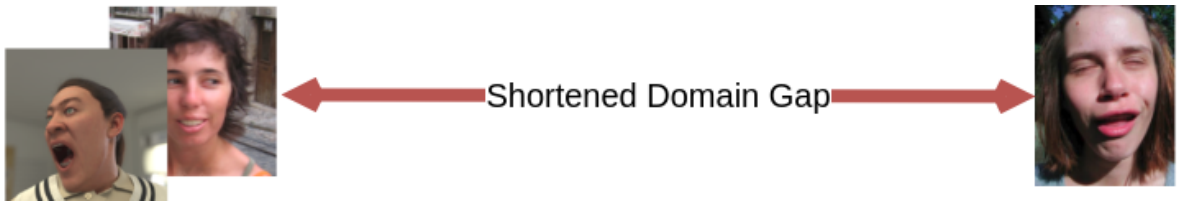


Figure 11: Shortening of the domain gap between iBugMask's test set and the Domain-Mixed dataset, via its inclusion of highly expressive synthetic data and real-face data.

Returning to the third aim outlined in Chapter 1.1, I have also determined the effects that mix-

ing real and synthetic data has on in-the-wild face parsing performance. If used alongside real-face data, either via domain-mixing in the initial model training or later via the training of a label adapter, supported by the “Mixed(50:50)” and post-adaptation “Synthetic” results from Table: 7, then synthetic-based training data perform extraordinarily well and can actually outperform purely real-face alternatives as shown in Table: 7.

4.2 Label Adaptation Results

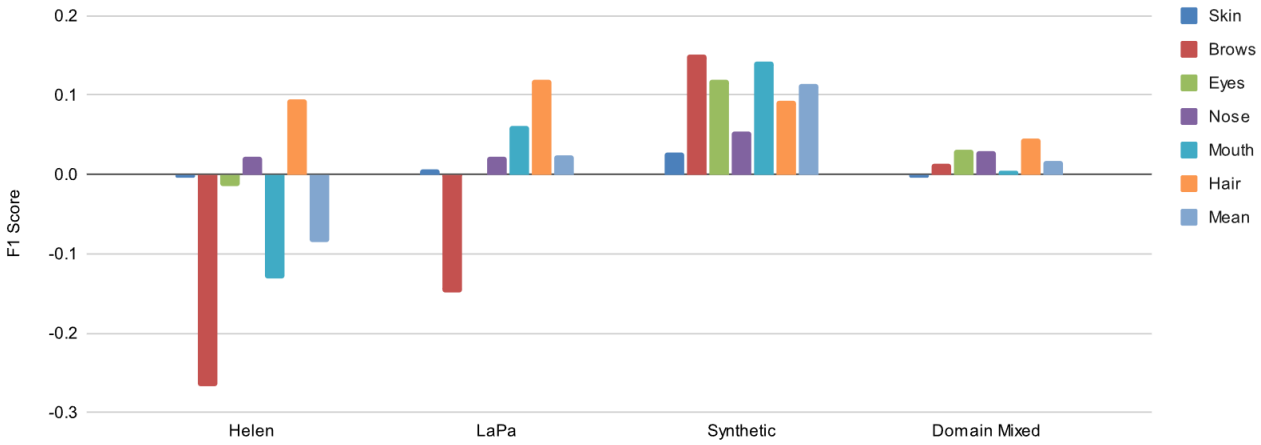


Figure 12: Plotting of the average difference in F1 scores of key facial categories before and after label adaptation from DeepLabV3+, FPN, UNet, and MobileNetV2 on iBugMask’s test set.

Similar to what was described in Wood et al.’s paper, label adaptation, specifically in the purely synthetically-trained models, not only saw a massive recovery of described semantic components but also managed to learn and repair inter-class patterns such as the chin (see Figure: 13). Furthermore, dense recovery of semantic components was displayed in classes both large and small. This can be seen in the top of Figure: 13, where the skin - originally misclassified as “background”, is corrected to the skin class, and the chin and lower lip are recovered to a high degree of accuracy. However, in the real-face datasets, label adaptation either made either very slight improvements or actually decreased the final prediction’s accuracy. After label adaptation, particularly in the LaPa-trained models, the models forgot how to predict the left eye. This is reflected in the NaN values displayed in Table: 7 and can also be seen in Figure: 14. Significant deterioration in prediction accuracy post-adaptation can also be seen in the “Brows” category in both the LaPa and Helen trained models in Figure: 12. This is most likely due to there being no obvious learnable pattern between them and iBugMask’s test set’s images that can be learned within the 30 epochs I limited the adapters’ training to.

Returning to the second aim outlined in Chapter 1.1 of providing the community with results pre- and post-label adaptation and sharing insight on how important this stage is for synthetic and real-face data, it can be concluded from my results that label adaptation is a crucial role in achieving competitive performance between the synthetic data provided by Wood et al. and real-face data such as Helen. The magnitude of the effect label adaptation had on each set of models can be seen in Figure: 10, in which the most significant increase in F1 can be seen in the synthetic dataset. This is most likely, as described in Wood et al.’s paper, due to the label adapter helping address the systematic difference present between labels generated by a computer, and labels annotated by hand. Thus, the datasets with a higher proportion of synthetic data benefited more.

From Table: 7, post-adaptation, the models trained with synthetic data performed the highest when transferred to an in-the-wild environment. The best results post-adaptation were: the domain mixed dataset in 1st, averaging 0.611; the upper bound in 2nd with 0.610; the fully synthetic dataset in 3rd, averaging 0.598; LaPa in 4th, averaging 0.560; and Helen in last place, averaging 0.500. The domain-

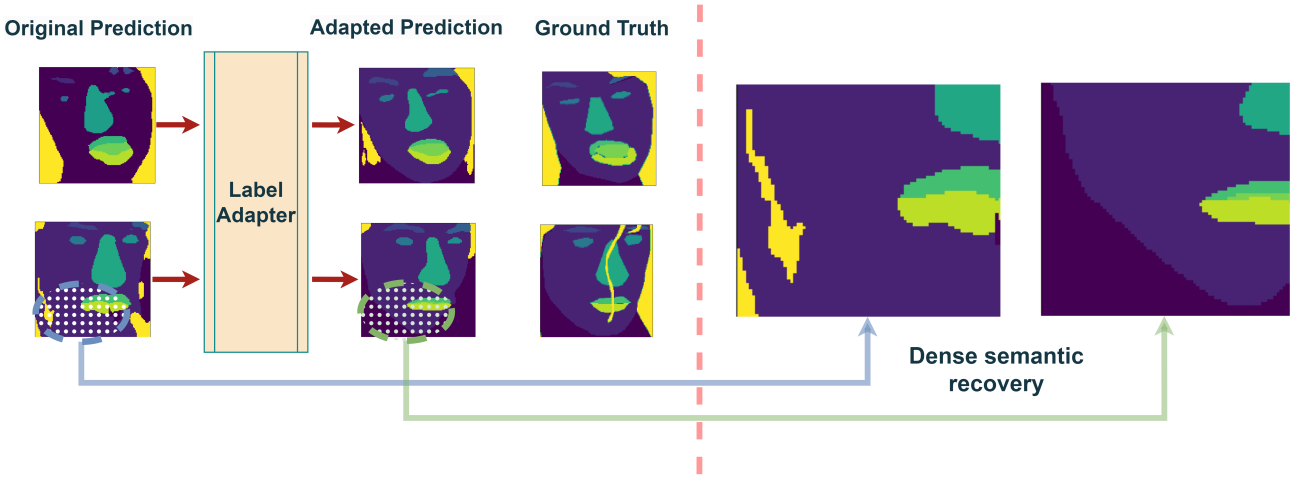


Figure 13: Dense recovery of semantic components post-adaptation from the synthetically trained models.

mixed and fully synthetic dataset outperformed Helen and LaPa within most semantic categories. However, despite Microsoft's synthetic data performing the best post-adaptation, the pre-adaptation results that were omitted from Wood et al.'s paper tell a completely different story. Pre-adaptation, the best datasets were Lapa in first with an average f1 of 0.698, the domain-mixed dataset in second with 0.594, Helen in third with 0.585, and Microsoft's fully synthetic dataset actually in last place with 0.483. From this, it is evident why Microsoft did not include the pre-adaptation results in their paper. Because the label adaptation stage involves the use of real images, these results prove that the use of real-face images was imperative in achieving their competitive results, and without it, their results are far worse than both of the real-face datasets they compared against. This greatly contrasts the idea conveyed throughout their report and even in their subtitle: "Face analysis in the wild using synthetic data *alone*".

4.3 Different Models' Results

Dataset	Model	Skin	Brows	Eyes	Nose	U-Lip	I-Mouth	L-Lip	Hair	Mean
Helen	**MobileNetV2**	0.914	0.461	0.519	0.858	0.619	0.552	0.675	0.561	0.614
LaPa	**MobileNetV2**	0.920	0.461	0.519	0.878	0.627	0.670	0.723	0.678	0.631
Synthetic	**LA MobileNetV2**	0.901	0.482	0.496	0.863	0.592	0.659	0.679	0.536	0.618
Domain Mixed	**LA MobileNetV2**	0.899	0.475	0.523	0.877	0.436	0.672	0.707	0.556	0.614

Table 8: The best models taken from each dataset's results. "LA" indicates that this is a post-adaptation result.

Returning to the fourth aim outlined in Chapter 1.1, from Figure: 8, we can see that all models performed relatively similarly in this task across most semantic categories. All models, regardless of training data, performed very highly in the task of face-parsing in the wild. However, from Table: 8, in contrast to my hypothesis in Chapter 3.1 of FPN performing the highest due to its inclusion of Feature Pyramids, MobileNetV2 was the best model for each training set. As such, the MobileNetV2 encoder with Unet-style decoder prevails in this research as the best model for face parsing.

Using the coordinates previously used for faced alignment in the prepossessing of the image, the colour-coded segmentation map was then overlaid back onto the original image using OpenCV. From this, we can see the segmentation as it appears in the wild, and better understand disparities between model performances. A particular difference that surfaced from this, was a particularly low performance in the predictions of eyes and brows from DeepLabV3+ and UNet - an error more frequent pre-adaptation (see Figure 14). Furthermore, these models often predicted these components in non-focus subjects within the given image (see Figure 14). suggesting that they could not learn that there should only be two eyebrows and two eyes per prediction. FPN is most likely immune to this problem due to the feature pyramids helping prediction accuracy of classes at smaller scales like the eyes and eyebrows, however, it is unclear as to why MobileNetV2 is also immune - a question I

open up to further research.

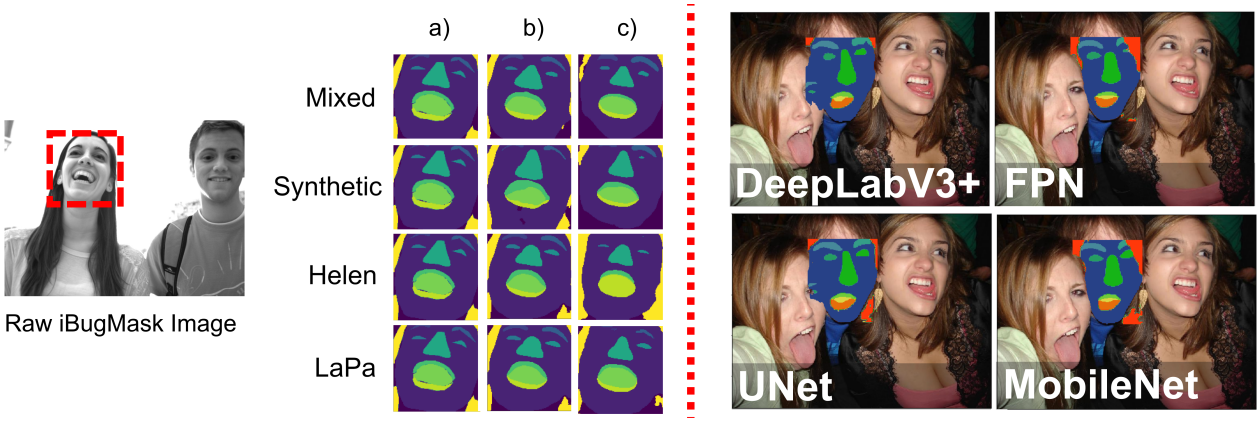


Figure 14: Left: Using MobileNetV2, a) Ground Truth, b) pre-adaptation, c) post-adaptation. Right: in the wild segments from the pre-adapted Helen-trained models.

4.4 Quality vs Quantity

Returning to the fifth aim outlined in Chapter 1.1 regarding the quality of the training data provided by Wood et al., It can be convincingly determined that after adaptation, Wood et al.'s dataset is more valuable Helen, and on par with LaPa. Wood et al.'s synthetic dataset outperformed a real-face dataset of the same size (Helen) and achieved almost identical performance with a dataset more than 10x its size (LaPa). In Wood et al.'s research, they used a 100,000-sized dataset for the synthetically-trained models and a 2,000 for the real-face trained models (Helen-trained), so a direct comparison of the value of each dataset's training material could not be made. However, in my research, a direct value comparison can be calculated from the percentage difference between Helen's and Microsoft's training data. Calculated from the average results post-adaptation, the synthetic data provided by Wood et al. is 17.9% better than Helen. Pre-adaptation, however, Wood et al.'s dataset is 19.1% worse than Helen.

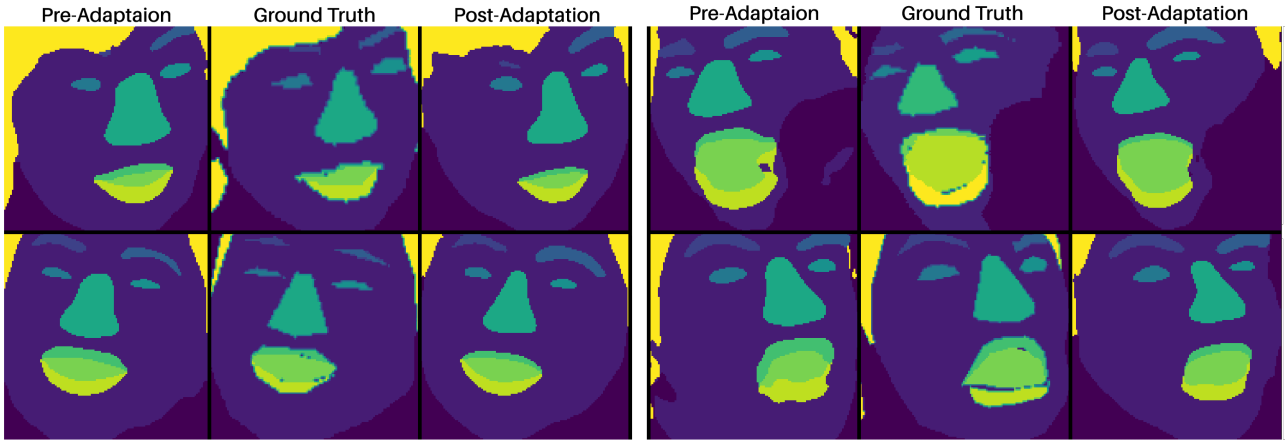


Figure 15: Domain-mixed MobileNetV2 Predictions of images taken from iBugMask's test set.

5 Project Discussion and Conclusion

Reflecting on the results of my research, I do not find it surprising that the datasets using synthetic images performed so well, as this can be deduced from the findings of Wood et al. What is surprising, however, due to its omittance from Wood et al.'s paper, is that the models trained *purely* on synthetic data performed far worse than any models trained with real-face data. This highly suggests the state-of-the-art of synthetic faces for face parsing is not yet at a point where it can replace real-face data, but that it is a powerful tool to increase diversity and thus final prediction accuracy when used alongside real data. The magnitude of the effect label adaptation had on the synthetic data is similarly

shocking, and it can be concluded that the synthetic data provided by Wood et al. is not worth using without it. Another surprising outcome from my research is how similar all models designs performed, regardless of the training data. As highlighted in Chapter: 3.1, FPN was expected to outperform all other models due to its increased ability to capture semantic components of disparate sizes within an image. Despite this, all models performed very similarly, with MobileNetV2 providing the best architecture for in-the-wild face parsing.

Reflecting upon my research, I believe I have achieved all of the aims outlined at the start of this project and followed all points of the experiment's success criteria. I have provided the community with a detailed evaluation of the performance of synthetic and real data in the task of face parsing in a truly in-the-wild environment; a detailed understanding of how and what particular datasets label adapting makes the most difference; discovered which model performed the best in the task of face parsing in the wild; provided a comparison on the value of the synthetic vs real-face data; and provided the community with open-source code on all experiments conducted.

However, there are a few topics that I could not discuss in this paper, whose results would be beneficial to this research area. Why the MobileNetV2 with Unet-style decoder model performed the best is unclear, as its architecture is far less complex and suited to this domain area than other models such as FPN. Understanding why this model performed best could help in the designing of better face parsing models in the future, as such I open this topic to the community for further research. Furthermore, the effects of different model architectures to be used as label adapters would be another interesting area for further research, as well as what effect intuitively obtaining the coordinates used to crop the image around the RoIs have when tested in the wild has. Because of this, I have included the code both for an FCN label adapter and Haar Cascade face-aligner in the code repo for use in further research. Finally, I would have also liked to repeat these experiments with more computing power, to allow for more epochs during initial model training and adapter training, allowing me to push the accuracy of these models even further. With more training time, more optimal hyperparameters and augmentation combinations could have been determined easier.

In conclusion, it is evident that state-of-the-art synthetic data can not only compete but actually outperform real-face alternatives. The rewards of synthesizing training data, such as more challenging and diverse images, can most certainly be reaped in the task of face parsing in the wild. However, a crucial point that fundamentally contrasts the message conveyed in Wood et al.'s paper is that the inclusion of real-face data for use alongside synthetic data is not only imperative but required in order to gain competitive results to real-face datasets like Helen and LaPa. Furthermore, Although Wood et al.s paper omitted the pre-adaptation results from their research, my inclusion of it revealed how poorly the synthetic data performed without adaption in contrast to the other datasets. As such, it can be concluded that when used alongside real data, the state-of-the-art of synthetic face images for face parsing is incredibly effective, and I recommend its use in the task of face parsing in the wild to the community, and I would love to see it being used in future face-parsing research. When used alone, however, the status quo remains unscathed, and the novel dataset released by Wood et al. remains inferior to real-face alternatives like Helen and LaPa.

References

[1] Ivona Tautkute, Tomasz Trzcinski, and Adam Bielski. "I know how you feel: Emotion recognition with facial landmarks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 1878–1880.

[2] Jisoo Park et al. "An automatic virtual makeup scheme based on personal color analysis". In: *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*. 2018, pp. 1–7.

[3] Aniwat Juhong and Chuchart Pintavirooj. “Face recognition based on facial landmark detection”. In: *2017 10th Biomedical Engineering International Conference (BMEiCON)*. IEEE. 2017, pp. 1–4.

[4] Erroll Wood et al. “Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 3681–3691.

[5] Jonathan Tremblay et al. “Training deep networks with synthetic data: Bridging the reality gap by domain randomization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 969–977.

[6] Jakob Geyer et al. *A2D2: Audi Autonomous Driving Dataset*. 2020. DOI: 10.48550/ARXIV.2004.06320. URL: https://arxiv.org/abs/2004.06320.

[7] Kimmo Karkkainen and Jungseock Joo. “FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 1548–1558.

[8] D. J. Butler et al. “A naturalistic open source movie for optical flow evaluation”. In: *European Conf. on Computer Vision (ECCV)*. Part IV, LNCS 7577. Springer-Verlag, Oct. 2012, pp. 611–625.

[9] Ankur Handa et al. *SceneNet: Understanding Real World Indoor Scenes With Synthetic Data*. 2015. DOI: 10.48550/ARXIV.1511.07041. URL: https://arxiv.org/abs/1511.07041.

[10] Philipp Fischer et al. *FlowNet: Learning Optical Flow with Convolutional Networks*. 2015. DOI: 10.48550/ARXIV.1504.06852. URL: https://arxiv.org/abs/1504.06852.

[11] Nikolaus Mayer et al. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: 10.1109/cvpr.2016.438. URL: https://doi.org/10.1109%2Fcvpr.2016.438.

[12] Weichao Qiu and Alan Yuille. *UnrealCV: Connecting Computer Vision to Unreal Engine*. 2016. DOI: 10.48550/ARXIV.1609.01326. URL: https://arxiv.org/abs/1609.01326.

[13] Yi Zhang et al. “UnrealStereo: A Synthetic Dataset for Analyzing Stereo Vision”. In: (Dec. 2016).

[14] John McCormac et al. *SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth*. 2016. DOI: 10.48550/ARXIV.1612.05079. URL: https://arxiv.org/abs/1612.05079.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[16] Tianchu Guo et al. “Residual Encoder Decoder Network and Adaptive Prior for Face Parsing”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.12268. URL: https://ojs.aaai.org/index.php/AAAI/article/view/12268.

[17] Jinpeng Lin et al. *Face Parsing with RoI Tanh-Warping*. 2019. DOI: 10.48550/ARXIV.1906.01342. URL: https://arxiv.org/abs/1906.01342.

[18] Vuong Le et al. “Interactive Facial Feature Localization”. In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 679–692. ISBN: 978-3-642-33712-3.

[19] Yinglu Liu et al. “A New Dataset and Boundary-Attention Semantic Segmentation for Face Parsing”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (Apr. 2020), pp. 11637–11644. DOI: 10.1609/aaai.v34i07.6832. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6832.

[20] Yiming Lin et al. *RoI Tanh-polar Transformer Network for Face Parsing in the Wild*. 2021. DOI: 10.48550/ARXIV.2102.02717. URL: https://arxiv.org/abs/2102.02717.

[21] Xiaoxing Zeng, Xiaojiang Peng, and Yu Qiao. “DF2Net: A Dense-Fine-Finer Network for Detailed 3D Face Reconstruction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[22] Haibo Qiu et al. *SynFace: Face Recognition with Synthetic Data*. 2021. DOI: 10.48550/ARXIV.2108.07960. URL: https://arxiv.org/abs/2108.07960.

[23] Liang-Chieh Chen et al. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018. DOI: 10.48550/ARXIV.1802.02611. URL: https://arxiv.org/abs/1802.02611.

[24] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2016. DOI: 10.48550/ARXIV.1612.03144. URL: https://arxiv.org/abs/1612.03144.

[25] Andrew Howard et al. *Searching for MobileNetV3*. 2019. DOI: 10.48550/ARXIV.1905.02244. URL: https://arxiv.org/abs/1905.02244.

[26] Xinrong Hu et al. “Virtual try-on based on attention U-Net”. In: *The Visual Computer* 38.9 (2022), pp. 3365–3376.

[27] Ahana Roy Choudhury et al. “Segmentation of brain tumors using DeepLabv3+”. In: *International MICCAI Brainlesion Workshop*. Springer. 2018, pp. 154–167.

[28] Hongxing Peng et al. “Semantic segmentation of litchi branches using DeepLabV3+ model”. In: *IEEE Access* 8 (2020), pp. 164546–164555.

[29] Nur Suriza Syazwany, Ju-Hyeon Nam, and Sang-Chul Lee. “MM-BiFPN: multi-modality fusion network with Bi-FPN for MRI brain tumor segmentation”. In: *IEEE Access* 9 (2021), pp. 160708–160720.

[30] Yongzhe Yan et al. “Face parsing for mobile AR applications”. In: *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE. 2018, pp. 407–408.

[31] Humaid Alshamsi, Veton Kepuska, and Hongying Meng. “Real time automated facial expression recognition app development on smart phones”. In: *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2017, pp. 384–392. DOI: 10.1109/IEMCON.2017.8117150.

[32] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2016. DOI: 10.48550/ARXIV.1603.07285. URL: https://arxiv.org/abs/1603.07285.

[33] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: (2018). DOI: 10.48550/ARXIV.1801.04381. URL: https://arxiv.org/abs/1801.04381.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[35] Ozan Öztürk, Batuhan Sariturk, and Dursun Seker. “Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries”. In: *International Journal of Environment and Geoinformatics* 7 (Sept. 2020), pp. 272–279. DOI: 10.30897/ijegeo.737993.

[36] *Intuitive Explanation of Skip Connections in Deep Learning*. Accessed on October 16th 2022. 2022. URL: https://theaisummer.com/skip-connections/.

[37] Pavel Iakubovskii. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2019.

[38] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[39] Yinglu Liu et al. *A High-Efficiency Framework for Constructing Large-Scale Face Parsing Benchmark*. 2019. arXiv: 1905.04830 [cs.CV].

[40] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. “Do Better ImageNet Models Transfer Better?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[41] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[42] Brandon M Smith et al. “Exemplar-based face parsing”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 3484–3491.

[43] Alexander Buslaev et al. “Albumentations: Fast and Flexible Image Augmentations”. In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[44] Timothy Dozat. “Incorporating nesterov momentum into adam”. In: (2016).

[45] Yurii Evgen’evich Nesterov. “A method of solving a convex programming problem with convergence rate O\bigl(k^2\bigr)”. In: *Doklady Akademii Nauk*. Vol. 269. 3. Russian Academy of Sciences. 1983, pp. 543–547.

[46] Xiangxin Zhu and Deva Ramanan. “Face detection, pose estimation, and landmark localization in the wild”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 2879–2886.

[47] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. “Face alignment through subspace constrained mean-shifts”. In: *2009 IEEE 12th International Conference on Computer Vision* (2009), pp. 1034–1041.

[48] Ce Liu, Jenny Yuen, and Antonio Torralba. “Nonparametric Scene Parsing via Label Transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2368–2382. DOI: 10.1109/TPAMI.2011.131.

[49] Leon Gu and Takeo Kanade. “A Generative Shape Regularization Model for Robust Face Alignment”. In: Oct. 2008, pp. 413–426. ISBN: 978-3-540-88681-5. DOI: 10.1007/978-3-540-88682-2_32.

[50] Yinglin Zheng et al. *General Facial Representation Learning in a Visual-Linguistic Manner*. 2021. DOI: 10.48550/ARXIV.2112.03109. URL: https://arxiv.org/abs/2112.03109.

[51] abhilashgaurav003. 2023. URL: https://www.geeksforgeeks.org/face-alignment-with-opencv-and-python/.