CMPT 732 - Project Report
Bridging the Domain Gap

Alexander Mountain - 301553106

Sridhar Suresh Ragupathi - 301456071

Traven Blaney - 301539045

GITHUB LINK: https://github.com/ajdm432/bridging_domain_gap

TABLE OF CONTENTS

Problem Statement

The problem statement for this work was provided by Industrial Light & Magic (ILM). The goal was to design and train facial landmarks models on the provided Fake It Till You Make It (Fake-It) dataset from Microsoft (Wood et al., 2021), and to test the models on real world faces. These tasks were to be completed so that results could be efficiently computed on consumer-level GPUs. Two additional bonus tasks were also completed: running the best model in real time over webcam feed (30 fps) and identifying which data augmentation techniques offered the most improvements to model accuracy for this problem.

Datasets

Training: Fake it till you make it

As a training dataset, the Fake It Till You Make It dataset from Microsoft was used, which contains 100,000 images of computer generated faces along with their facial landmark annotations. These include 70 keypoints: 68 from the standard format used in works like the 300W dataset (Sagonas et al., 2013), and 2 more for the pupil centers. The following experiments, use subsets of 1000, and 10,000 images from this dataset. This dataset contains good variations in terms of face orientation, pose, backgrounds and accessories on the faces.

Figure 1: Example images from the training dataset

Testing: 300W

Figure 2: Example images from the testing dataset

The 300W dataset was used for evaluating the chosen models. This is the most challenging dataset of its kind, containing 600 images with single or multiple faces in both indoor and outdoor settings. For the images with multiple faces, the keypoint annotation provided is for only one out of the faces. There are 2 variants of this dataset: the original version containing images with multiple faces and a cropped version, containing images with only one face which has been cropped according to the ground truth keypoint annotations. The cropped version was used in the following experiments as it is more similar to the training dataset, which also has

only 1 face per image. This version is also what comparable papers like Microsoft's Fake It Till You Make It used for testing.

Data Preprocessing and Augmentations

Since the models used were pre-trained on ImageNet, the images of the training data are normalized separately for each color channel using the ImageNet mean and standard deviations. The augmentations tested were translation, rotation, occlusion and blur.

All models included in this report were trained and tested using single-stage architectures. State-of-the-art methods in this domain typically employ a two-stage approach, whereby a face crop is performed on the given image prior to using the model - essentially removing the background. Face crops are typically made by any of the following methods: making a tight crop around the face from the facial landmark annotations (Jin et al., 2021), making a crop around the face based on a predefined bounding box (Kim et al., 2021), or using a pre-trained model to locate the face bounding box (Wood et al., 2021). This method of image augmentation was intentionally not performed here in order to evaluate the efficacy of training on purely synthetic images of human faces with backgrounds. For the purposes of this study, a two-stage approach would become reliant on the use of an external model trained on real human faces to locate the face bounding box during inference. Additionally, models using a two-stage approach would be less efficient, hindering one of the primary goals of this project; to perform inference on consumer GPUs at runtime.

Evaluation Metric - Normalized Mean Error

In order to compare our models with state-of-the-art models in the space, Normalized Mean Error was chosen as an evaluation metric. The NME calculation was normalized using interocular distance: the distance between the centers of pupils in each image. The last 2 keypoints provided in the training dataset keypoint annotations enable the calculation of this distance for each image.

Normalized Mean Error, generally expressed as a percentage (%) is given by:

$$NME(S,S^*) = \frac{1}{N}\sum_{i=1}^{N}\frac{\| s_i - s_i^* \|_2^2}{L}$$

Where S,S* correspond to the ground truth and predicted keypoints respectively and L is the normalization factor (interocular distance in this case). The lower the NME, the better a given model is performing.

Architectures

ResNet50

ResNet50 is a 50 layer deep convolutional network with residual connections (He et al., 2015). It has been shown to perform with high accuracy on image problems such as segmentation and object detection. One flaw with such a deep network is that it requires larger amounts of data and long training times to achieve these results. When training on Tesla T4s provided by Google, training times for the largest training dataset of 10,000 images could reach up to 12 hours. Theoretically, a benefit of ResNet50s depth is higher accuracy once it has been sufficiently trained. In this work the problem of large data requirements and long training times was partially mitigated by using a ResNet model which had been pre-trained on ImageNet. However, prohibitive training times and limited data still prevented this network from achieving results on a level comparable to the networks evaluated in the Fake It Till You Make It article.

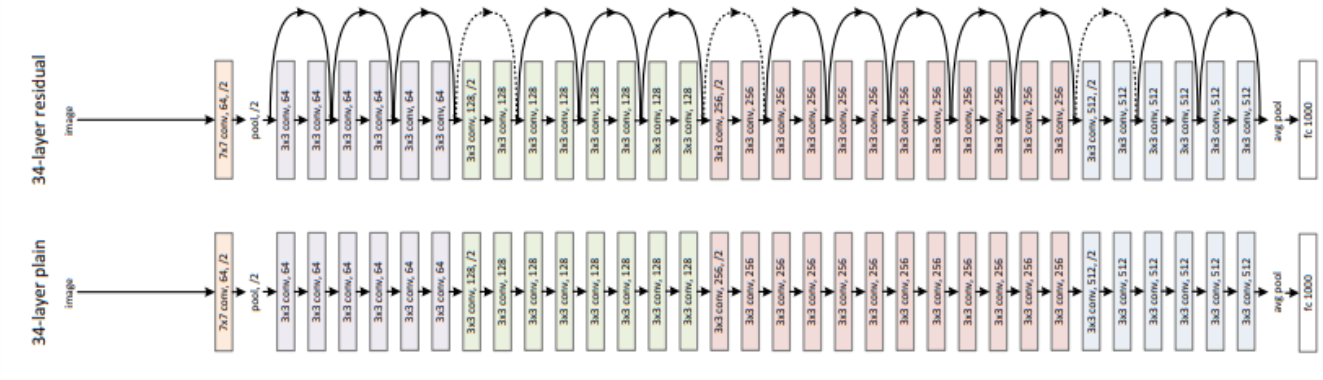


Figure 3: Plain deep convolutional model compared with a 34 layered ResNet architecture

MobileNetv3

The MobileNet line of architectures is aimed at reducing computation cost while minimizing reduction in accuracy (Howard et al., 2017). By using depth-wise separable convolution, Width multiplier for input channels and resolution multipliers, MobileNet and its variants are able to achieve a fast inference-speed. This model was chosen because it was expected to perform efficiently during real-time inference.

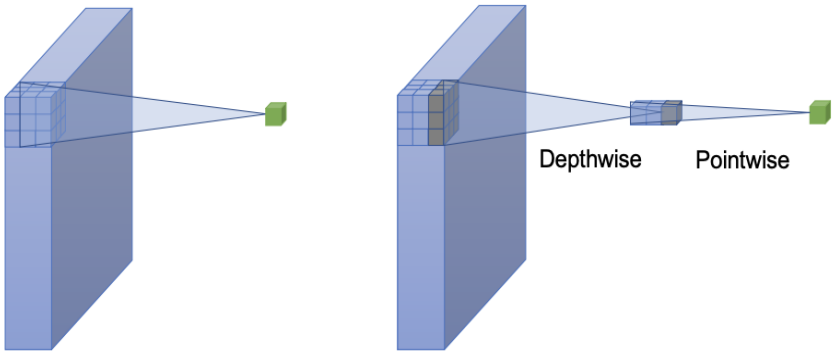


Figure 4: MobileNetv3 model architecture

Specifically, MobileNetv3 was used (Howard et al., 2019). This is the latest and best performing of the MobileNets in a direct regression approach where facial keypoints are predicted directly. MobileNetv3 was pre-trained on ImageNet as a backbone and the number of neurons in the last layer of the model was modified to be 68 to coincide with the 68 annotations provided in the test dataset. A transfer learning approach was used to train the network where the parameters of all but the newly added linear layer at the end are frozen.

PIPNet

Pixel-in-Pixel Net (PIPNet) is a model architecture that combines different categories of detection head - heatmap and direct coordinate regression - for the specific task of localizing facial landmarks. Heatmap regression is a form of detection head used to perform facial landmark localization that has been modified from similar image segmentation tasks. Models that use this form of detection head map an image to high-resolution heatmaps, each of which correspond to the probability of a facial landmark location. While this method is more computationally expensive than its direct regression counterpart, due to the need to upsample featuremaps back to the input image dimensions, heatmap regression is able to reach state-of-the-art level accuracy with single-stage approaches - whereas direct regression typically requires a two-stage approach to attain similar results (Jin et al., 2021). In general, heatmap regression is more accurate in terms of globalization, meaning it is able to locate a region of interest over the entire image more precisely due to the additional information passed through to the model in the form of heatmaps (Earp et al., 202). At the same time, coordinate regression is typically more accurate at localisation, meaning it is better suited to inferring actual landmark coordinates once a region of interest has been identified (Earp et al., 2021). PIPNet attempts to merge the best qualities from both types of regression to create a model that is both accurate and efficient, by conducting heatmap regression globally and coordinate regression locally (Jin et al., 2021).

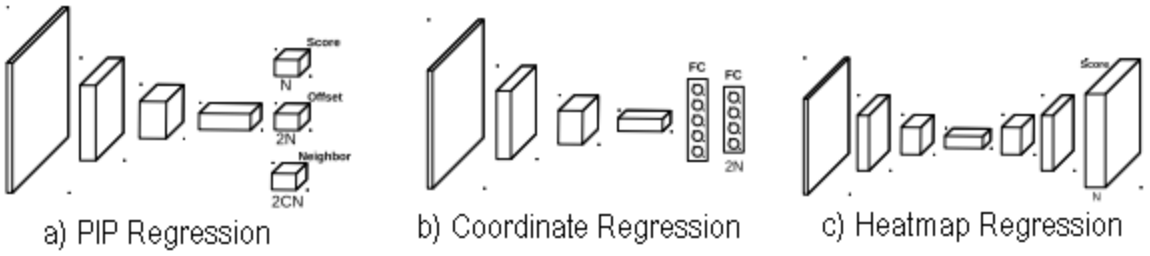
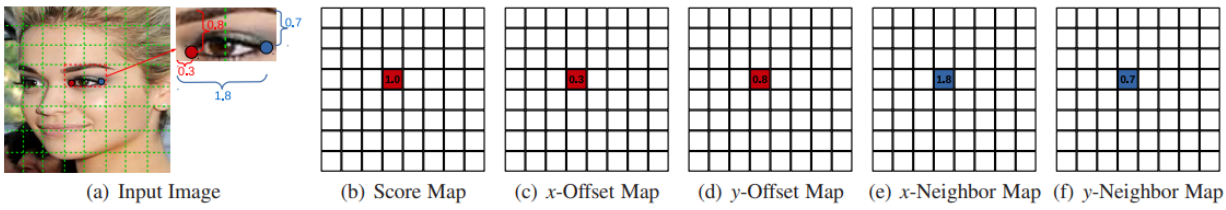


Figure 5: PIPNet model architecture, as well as example architectures for typical heatmap and coordinate-based regression methods (Jin et al., 2021)

Unlike other purely heatmap-based regression approaches, PIPNet argues that low resolution feature maps are sufficient for landmark localization. The accuracy is furthermore increased through predicting offsets in the x and y direction within each heatmap grid, so as to not lose sub-pixel information from the low-resolution score map - a common issue in straightforward heatmap regression (Jin et al., 2021). Moreover, in order to improve robustness, “PIP regression” also predicts the offsets of *C* neighboring landmarks, where *C* is an editable hyperparameter (Jin et al., 2021). The final predictions are an average of both the values from each landmark’s score map (including the x, y offset maps), and their offset values from neighboring predicted points.



(a) A sample image as input. The red dot denotes the target ground-truth landmark, and the blue one is a neighboring landmark.
(b) Label assignment for the score map. (c)-(d) Label assignment for the offset maps on x and y axes, respectively.
(e)-(f) Label assignment for the neighbor maps on x and y axes, respectively.

Figure 6: Mapping a ground-truth landmark label to heatmap labels in PIPNet form

Figure 6 demonstrates what a converted ground truth landmark label would look like once converted into heatmaps. The net stride of the network is an editable hyperparameter, which controls the dimensions of the final prediction score maps. As is the case in Figure 6, with an input image size of 256x256 pixels, and a net stride of 32, the resulting low-resolution feature map has dimensions 8x8 pixels. For example, a net stride of 16 would result in a low-resolution feature map size of 16x16 for the same input image dimension. For the purposes of this report, a net stride of 32 was selected as it provided the lowest NME values when trained and tested on the 300W dataset in the original PIPNet article (Jin et al., 2021). Similarly, a pretrained ResNet18 backbone was selected as it provided the best efficiency-accuracy tradeoff when trained and tested on the 300W dataset in the original PIPNet article (Jin et al., 2021).

Experiments

Two different subsets of training data were used to assess the minimum viable dataset size for crossing the domain gap: 1,000 and 10,000 images.

Differing combinations of augmentation techniques were used to evaluate the effect these augmentations play in the ability for the model to learn. The models trained with no augmentations at all acted as a baseline. After initially testing all of the models with and without all augmentations included in this project, it was noted that for MobileNetv3 and ResNet50, the model version trained on augmented images resulted in worse (greater) NME values during testing. To explore this further a third subset of augmentations was introduced. This included just occlusion and translation, as it was hypothesized that the training data already showed sufficient examples with rotation, and blurring was not an overly relevant augmentation. Moreover, coordinate regression-based-methods using a single-step approach typically suffer from poor global accuracy - additional rotation might have confused these models more than helped them. These three subsets were evaluated and compared based on their results.

In total, there are 3 different data augmentation combinations: no augmentation, translation and occlusion only, and all augmentations (rotation, blur, translation and occlusion). This led to a total of 6 experiments for each model, the results of which are discussed in the next section.

The following figures present plots of training, validation losses and validation NME values for the experiments summarized above.

ResNet50

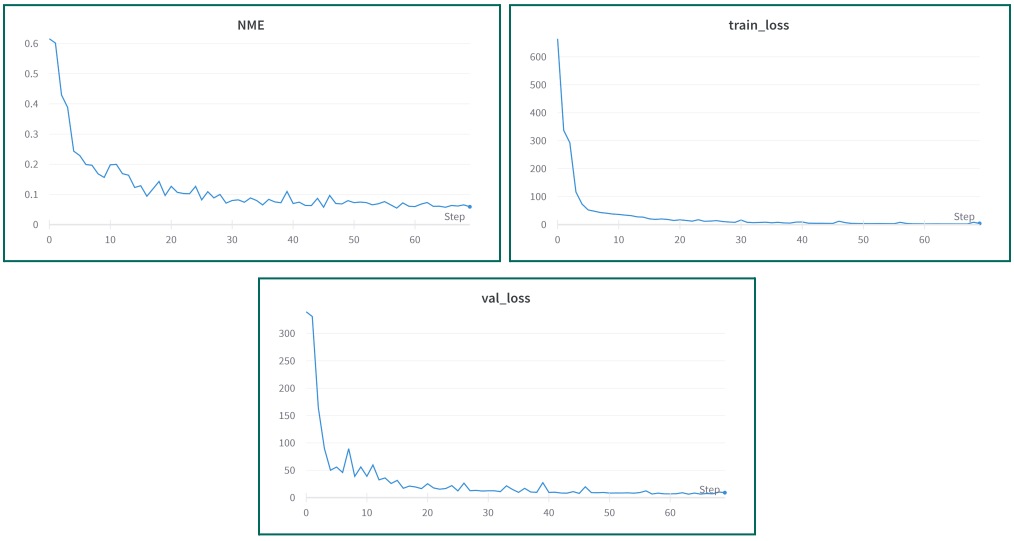


Figure 7: Validation NME values (top left), training loss values (top right), validation loss values (center bottom) for ResNet50

MobileNetv3

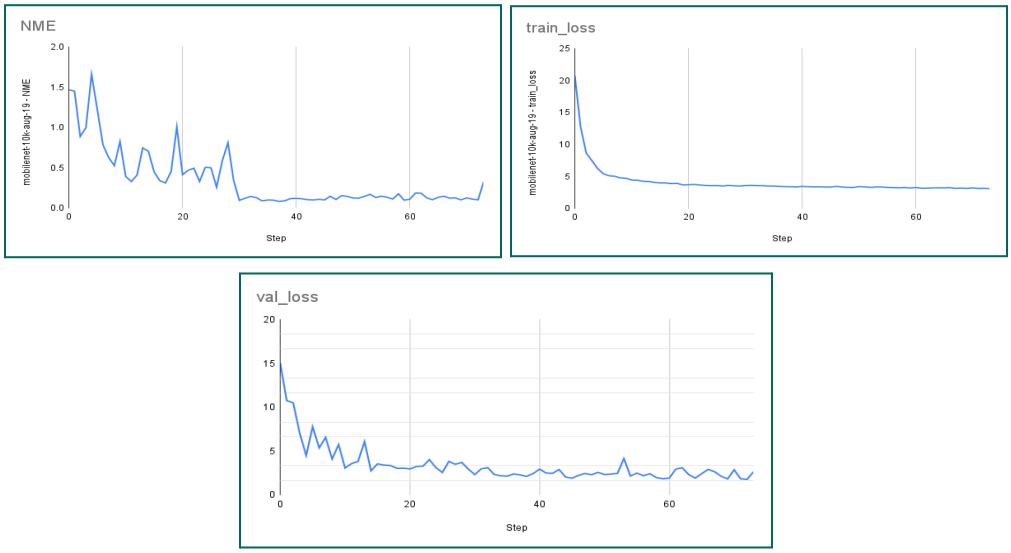


Figure 8: Validation NME values (top left), training loss values (top right), validation loss values (center bottom) for MobileNetv3

PIPNet

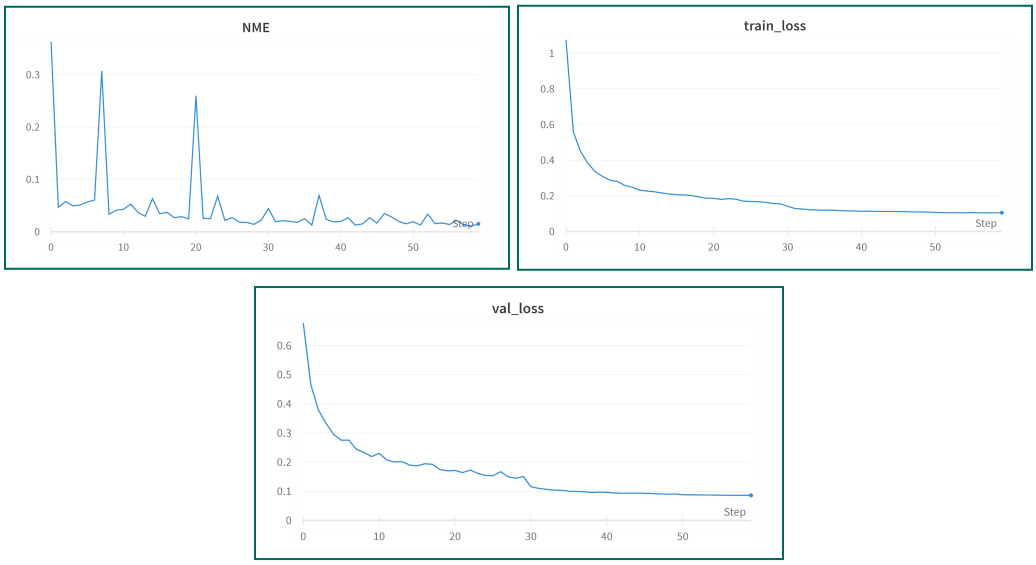


Figure 9: Validation NME values (top left), training loss values (top right), validation loss values (center bottom) for PIPNet

Results

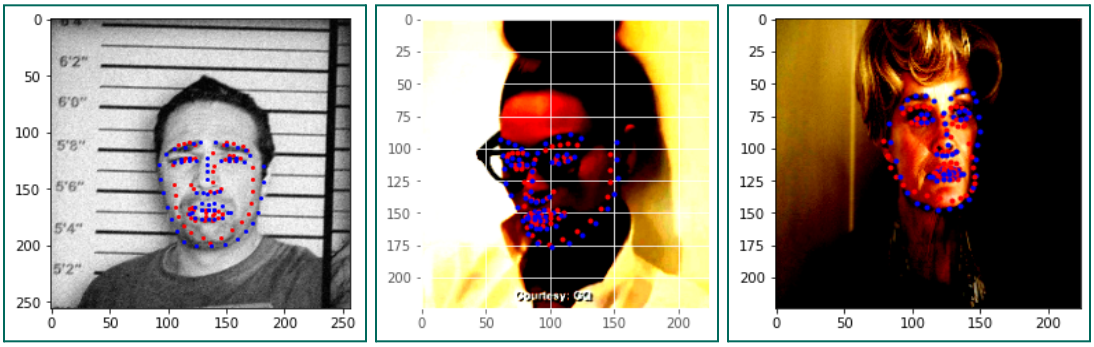


Figure 10: Test set example results from PIPNet (left), ResNet50 (middle), and MobileNetv3 (right) showing lower NME values.

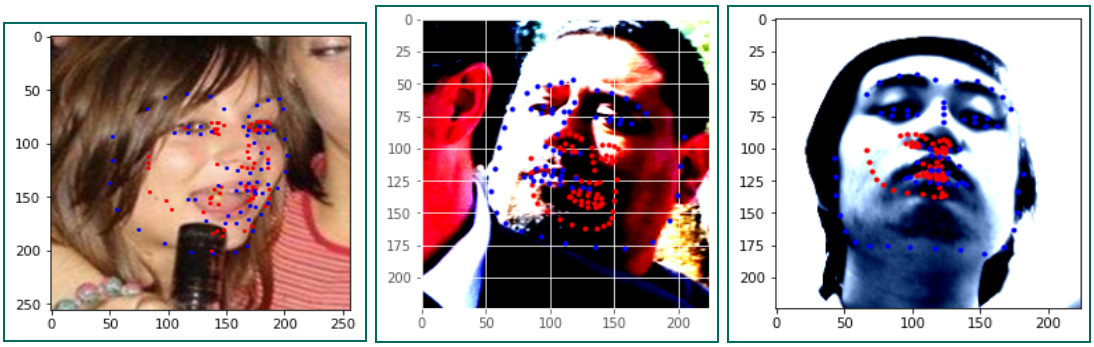


Figure 11: Test set example results from PIPNet (left), ResNet50 (middle), and MobileNetv3 (right) showing higher NME values.

Training Information for Best Models

Table 1: Training hyperparameters for best-case models

	ResNet50	PIPNet	MobileNetv3
Epochs	70	60	70
Learning Rate	0.001	0.0001	0.001
Batch Size	8	16	32
Loss Function	MSE	L2 and L1	MSE
Inference Time	530 ms/image	11.2 ms/image	7.16 ms/image

Input dimensions	244x244	256x256	244x244
Net Stride (PIPNet)	NA	32	NA
Neighbor Amount (PIPNet)	NA	10	NA

Inference time was compared using Google Colab GPUs so all measurements were performed on Tesla T4s. However possible inconsistencies may have arisen due to varying resource allocation by Colab. The probability of such an issue seems high since despite its slow performance here, ResNet50 ran at 30 FPS on a local machine when testing webcam integration.

Normalized Mean Error

Table 2: Results on the 1,000 image training data subset

	ResNet50	PIPNet	MobileNetv3
No Augmentation	79	14.4	71.3
Translation and Occlusion	58.3	13.8	45.43
All Augmentations	64.7	13.1	60.05

Table 3: Results on the 10,000 image training data subset

	ResNet50	PIPNet	MobileNetv3
No Augmentation	32	16.1	23.9
Translation and Occlusion	44	14.5	42.1
All Augmentations	58.98	13.9	60.9

Graph Representations

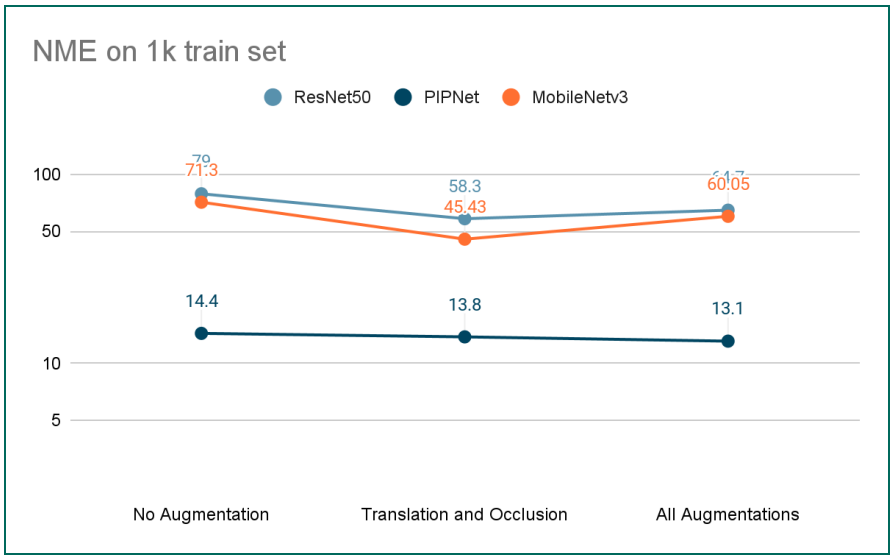


Figure 12: Graph demonstrating NME values for the different augmentation techniques for the 1,000 image training subset

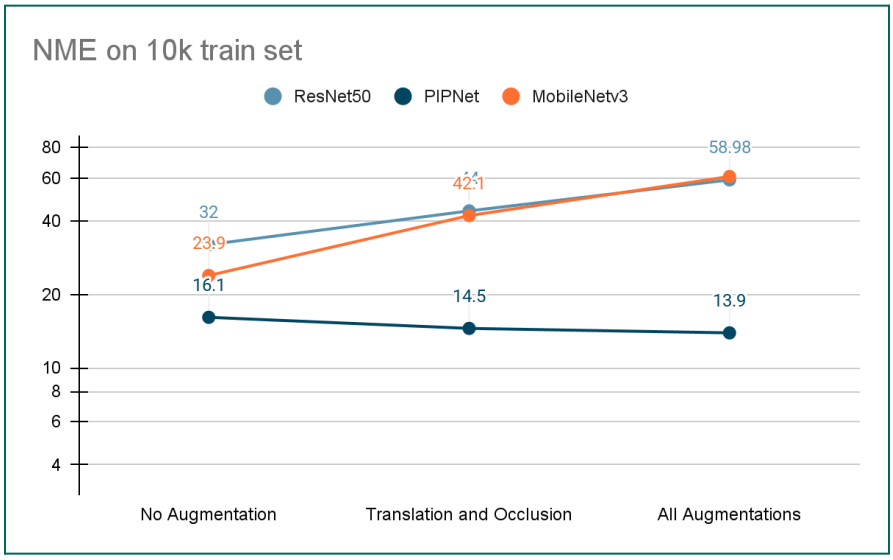


Figure 13: Graph demonstrating NME values for the different augmentation techniques for the 10,000 image training subset

Generally, networks performed better when trained on the larger dataset, although PIPNet performed better after being trained on just 1,000 images. Networks trained on just 1,000 images benefited to a greater degree from augmentations, while the networks trained on 10,000 generally did not, with PIPNet again being an exception. Results improve on the 1,000 image trained networks with just translation and occlusion, which is believed to add more variation, and therefore information, to the dataset, improving the learning capability of the model. From visual inspection, it is evident that there are already a large amount of images demonstrating variation in rotation in the training dataset. While augmentation generally improved PIPNet's performance, both ResNet50 and MobileNetv3 did somewhat poorly when

learning from augmented data.

Average Inference Time

Table 4: Average inference time per image for each model

Model	Inference Time (ms/image)
MobileNetv3	7.16
PIPNet	11.2
ResNet50	530

Average Inference time is computed as the time to predict facial landmarks for an image, averaged over the 600 images of the 300W test dataset. As expected, MobileNetv3 has the fastest inference time due to its lightweight architecture, followed by PIPNet. ResNet50 being a very deep architecture has the largest inference time of 530 ms per image on average.

Real-time inference using webcam

Each model was tested in real-time by running it over webcam footage. All models were capable of running in real-time on consumer-level GPUs at the benchmark of 30 FPS. However, model prediction accuracy and robustness varied. ResNet50 was capable of locating a face within the central area of an image with some accuracy, but failed when the face was near the edges of the screen. Furthermore, it was not robust to changes in background lighting. MobileNet and PIPNet both achieved accurate real-time results over webcam, although MobileNet was also not robust to changes in lighting. PIPNet made the most accurate predictions and was robust to changes in lighting, translation, rotation and scaling.

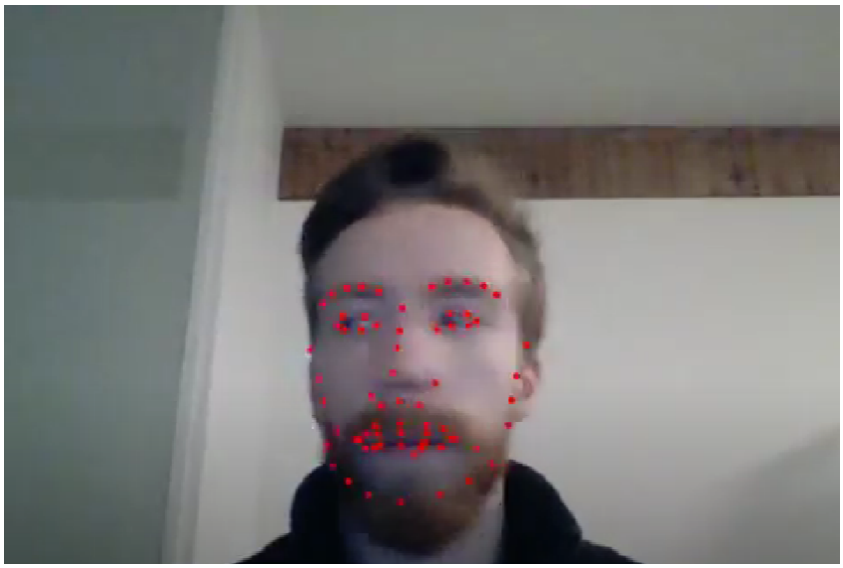


Figure 14: Real-time keypoint predictions
PIPNet (Inference on an NVIDIA GeForce 940M)

Figure 15: Real-time keypoint predictions
MobileNet (Inference on an NVIDIA GeForce GTX 1650 Ti Mobile)

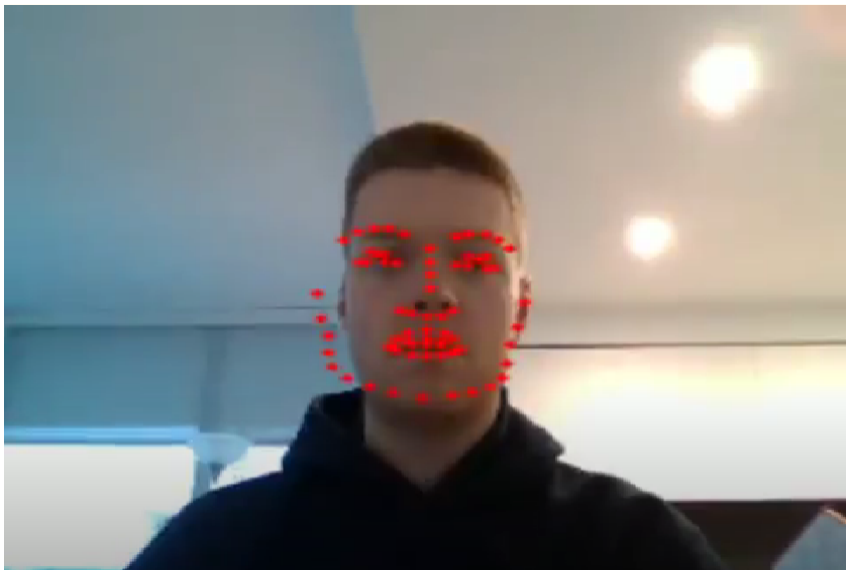


Figure 16: Real-time keypoint predictions
ResNet50 (Inference on an NVIDIA GeForce RTX 3050)

Interpreting Results

Best model: Low NME + Fast Inference Time

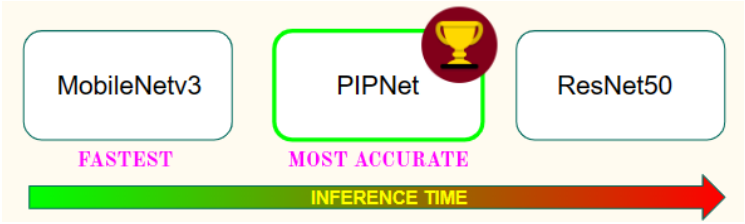


Figure 17: Graphic demonstrating project outcomes

While MobileNetv3 is the fastest at inference time, it is not the most accurate. PIPNet has only a slightly larger inference time than MobileNetv3 and is the most accurate both on the test data and on webcam inference. From the experiments conducted, it can be concluded that PIPNet is suitable for real-time facial landmark detection on consumer GPUs.

PIPNet likely outperformed MobileNetv3 and ResNet50 due to its primary form of detection head - heatmap regression. Because of the nature of this experiment following a single-step approach (no face cropping during training), models with a heatmap-based detection head were naturally better-suited to the problem. This is also in alignment with the differences in results between model architectures with varying augmentation techniques. With single-stage approaches image modifications such as rotation and translation have a significantly greater effect on coordinate-based detection heads than they do with heatmap-based detection heads.

Heatmaps provide more spatial information to the model than pure coordinates, resulting in greater globalization accuracies (finding the region of the key points over an entire image), while typically suffering low localisation accuracies (finding the key points when the face region is known). In the case of MobileNetv3 and ResNet50, both using coordinate-based detection heads, the models struggled locating the face globally over the image, especially when augmented further through rotation and translation, resulting in worse NME values. A two-stage approach with face crops would likely improve the results from both MobileNetv3 and ResNet50. PIPNet uses heatmap regression to essentially locate the region of interest with an image (the face), and then performs direct regression to find the coordinates within the region of interest.

PIPNet began to overfit to the synthetic data when trained on the 10,000 image dataset, which can likely be attributed to the limited data augmentation methods provided. As is evident in Figures 16 and 17, image augmentations helped limit the amount of overfitting when trained on the larger dataset. Due to the poor results with increased augmentations from the other networks, experimentation with a greater amount of augmentations was not warranted. However, it is hypothesized that PIPNet would continue to perform better with larger datasets by introducing more augmentations to training, as well as by introducing a decay factor to the learning rate.

Findings from Webcam Experimentation

MobileNetv3 and ResNet50 were found to perform poorly during webcam inference in dimly-lit environments, whereas PIPNet was more robust to changes in lighting in the models trained on augmented datasets. Furthermore, ResNet50 responded poorly to faces that were not in the central area of an image. As expected, coordinate-based detection heads (MobileNetv3, ResNet50) performed well in terms of maintaining the shape of the face (points look connected in a uniform way), which is due to its higher localisation accuracy, but struggled with rotational and translational changes. PIPNet, utilizing a heatmap-based detection head, was better at locating the facial region within each frame despite rotational and translational variations, but was less consistent in terms of maintaining accurate facial shapes.

Comparison with Other Work

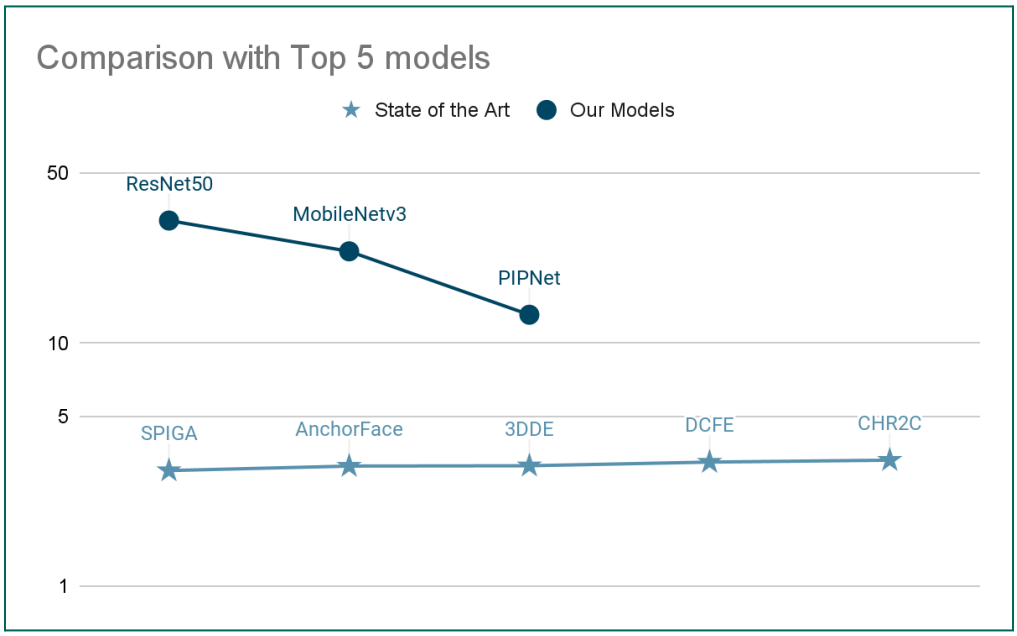


Figure 18: Comparison to the state-of-the-art facial landmark localization models

The above comparison shows NME values from the models tested in this paper in comparison with models tested in the Fake It Till You Make It paper. As demonstrated, there is still work to be done to achieve state-of-the-art results, even with PIPNet. This can likely be attributed to project limitations (see Limitations section). The original article which summarized the usage of the Fake It Till You Make It dataset to perform facial landmark localization did not achieve results similar (or better) in magnitude to the state-of-the-art approaches unless the model, in their case ResNet, trained on the full dataset of 100,000 images (Wood et al., 2021). This suggests that while training on synthetic data can achieve similar results to models trained on real-world images, a larger dataset of synthetic images is required to achieve equivalent results to a model trained on real-world images. This would likely be the case with PIPNet, provided a greater amount of augmentations were applied to the training dataset to mitigate overfitting to the synthetic data.

Future Scope

Potentially Useful Augmentation Techniques

The previous discussion of webcam inference has led to the idea of introducing a potentially useful data augmentation technique for MobileNetv3 and ResNet50: randomly varying the brightness of the image. Furthermore, ResNet50 would be served well by training on a dataset where the prevalence and magnitude of randomly translated images is increased. As

seen in the NME results, data augmentation worsened results for ResNet50, and it is likely that longer training times would be required for the network to see improvements from augmented data. Both ResNet50 and MobileNetv3 would benefit greatly from employing a two-stage approach during training, thus reducing the possible errors associated with translational and rotational bias in coordinate-based detection heads.

Having identified the most accurate network of those tested here, investing more time into data augmentation to increase the variation in the Fake-It dataset would be a critical next step. With a focus on improving the best network's performance, it is likely PIPNet could reach a level of accuracy closer to that of the networks tested in the Fake-It paper.

Applications

Once PIPNet achieves accuracy at the level of state of the art models, it would be worthwhile to integrate it with modern applications of keypoint detection. For example, generating synthetic images of people speaking from facial keypoints, as in the paper Few-Shot Adversarial Learning of Realistic Neural Talking Head Models (Zakharov et al., 2019).

Limitations

The results in this project were limited due to GPU access. Training was performed on Google Colab and on local machines, which meant limited access to powerful GPU resources. With greater access to such resources better results may have been seen especially for ResNet50, the deepest and most time-consuming network to train.

The model architectures were limited to single-stage approaches, which hindered the results of coordinate-based detection methods (MobileNetv3 and ResNet50) more acutely than that of heatmap-based detection methods (PIPNet). This was due to constraints with project parameters, specifically to ensure runtime inference on consumer GPUs, but also because of the untested nature of pretrained face bounding box localisation models on purely synthetic faces.

References

Earp, S. W. F., Samacoits, A., Jain, S., Noinongyao, P., & Boonpunmongkol, S. (2021). Sub-pixel face landmarks using heatmaps and a bag of tricks. *arXiv preprint*. arXiv:2103.03059

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778. https://doi.org/10.1109/cvpr.2016.90

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobilNetV3. *Proceedings of the IEEE/CVF international conference on computer vision*, 1314-1324. https://doi.org/10.48550/arXiv.1905.02244

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017, April 17). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *Cornell University*. https://doi.org/10.48550/arXiv.1704.04861

Jin, H., Liao, S., & Shao, L. (2021). Pixel-in-Pixel Net: Towards Efficient Facial Landmark Detection in the Wild. *International Journal of Computer Vision*, *129*(12), 3174-3194. https://doi.org/10.1007/s11263-021-01521-4

Kim, T., Mok, J., & Lee, E. (2021, August 9). Detecting Facial Region and Landmarks at Once via Deep Network. *Sensors (Basel)*. https://doi.org/10.3390/s21165360

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-Wild Challenge: The first facial landmark localization challenge. *2013 IEEE International Conference on Computer Vision Workshops*. 10.1109/iccvw.2013.59

Wood, E., Baltrusaitis, T., Hewitt, T., Dziadzio, S., Cashman, T. J., & Shotton, J. (2021). Fake it till you make it: Face analysis in the wild using Synthetic Data alone. *2021 IEEE/CVF*

International Conference on Computer Vision (ICCV).

https://doi.org/10.1109/iccv48922.2021.00366

Zakharov, E., Shysheya, A., Burkov, E., & Lempitsky, V. (2019). Few-Shot Adversarial Learning of Realistic Neural Talking Head Models. *Proceedings of the IEEE/CVF international conference on computer vision*, 9459-9468. https://doi.org/10.1109/iccv.2019.00955