Bridging The Domain-Gap: Facial Landmark Detection Using Synthetic Dataset

Simon Fraser University, CMPT 732, Fall 2022

Janani Nataraj
Student ID: 301561872
jna102@sfu.ca

Parth Gulati
Student ID: 301563893
pga56@sfu.ca

1. Introduction

This project aims to demonstrate that synthetic data may be used to train facial analysis algorithms before applying them in real-world circumstances, and that it is possible to perform face-related computer vision in the wild using just synthetic data.

Link for the project: https://csil-git1.cs.surrey.sfu.ca/pga56/cmpt-732-final-project

2. Methodology

Facial landmark points detection problem done by,

- Face detection using MTCNN
- Landmark detection using - ResNet18, ResNet50 and Xceptionet.

a) Dataset

We have used the Microsoft's synthetic dataset for training that combines a procedurally generated parametric 3D face model with a comprehensive library of hand-crafted assets to circumvent the issue of domain gap. These assets render diverse training images and high realism.

Purpose	Training and Validation	Testing
Dataset Used (Gray scale images)	Microsoft's Face Synthetics Dataset • 100k labeled images (used 10k) • 512*512 pixels (resized to 128*128 pixels) • 70 standard facial landmark annotations (used 68 landmark points)	Real Dataset • 300W for images • 300VW for video

Table 1: Datasets used in this project

b) Data Augmentations

Generating augmented images allows to have more and different data to train on without the need for collecting new labelled data. Data augmentations help with overfitting and enables to alter the data to better align with the intended application of the model.

- Photometric distortions - changing the contrast, brightness, saturation and hue
- geometric distortions - rotation and cropping

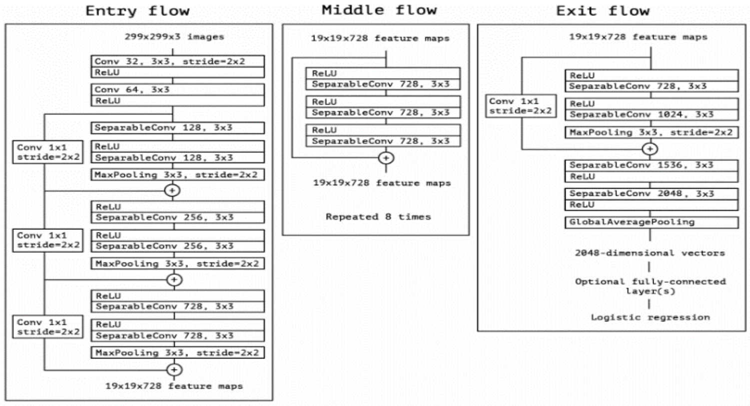
c) Models

i. Architecture

ResNet18 and ResNet50

layer name	output size	18-layer	34-layer	50-layer	101-layer
conv1	112×112	7×7, 64, stride 2			
		3×3 max pool, stride 2			
conv2_x	56×56	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax			
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9

Xceptionet



ii. Optimizer, Loss, Metrics

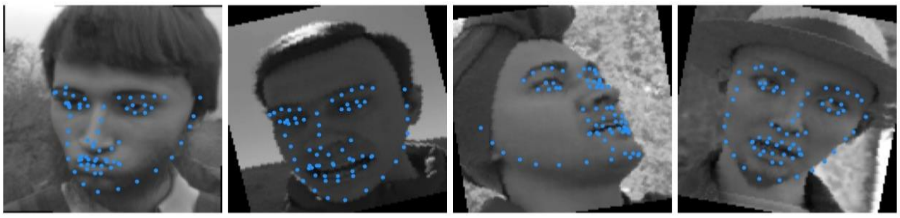
- Optimizer: pre-implemented Adam optimizer is used which is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
- Loss: Since we are dealing with specific distances between the predicted coordinates and the true coordinates, we are using the Root Mean Squared Error (RMSE)

iii. Training

Models	No of Images	Parameters	Time	Training Loss	Validation Loss
ResNet18	Total : 10k Validation split : 0.1	1. Learning Rate: 0.0001 2. No of Epochs: 3. Batch size: 32	173	0.02046739 (Epoch 83)	0.01785510 (Epoch 83)
ResNet50	Total : 10k Validation split : 0.1	1. Learning Rate: 0.0001 2. No of Epochs: 100 3. Batch size: 32	192	0.01687667 (Epoch 99)	0.01466818 (Epoch 99)
Xceptionet	Total : 10k Validation split : 0.1	1. Learning Rate: 0.0008 2. No of Epochs: 100 3. Batch size: 32	206	0.02138573 (Epoch 87)	0.01835696 (Epoch 87)

Table 2: Comparison of different models.

ResNet18

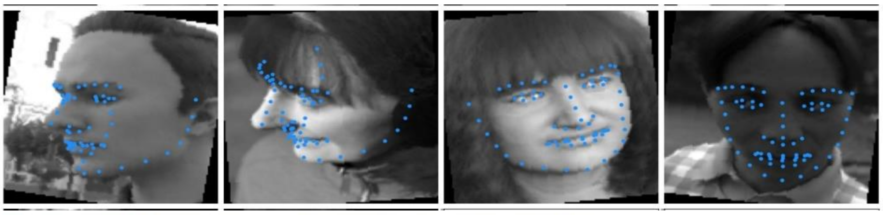


```
Minimum Validation Loss of 0.0179 at epoch 83/100
Model Saved

Epoch(83/100) -> Training Loss: 0.02046739 | Validation Loss: 0.01785510
```

Figure 1: Validation Images and the minimum validation loss (ResNet18)

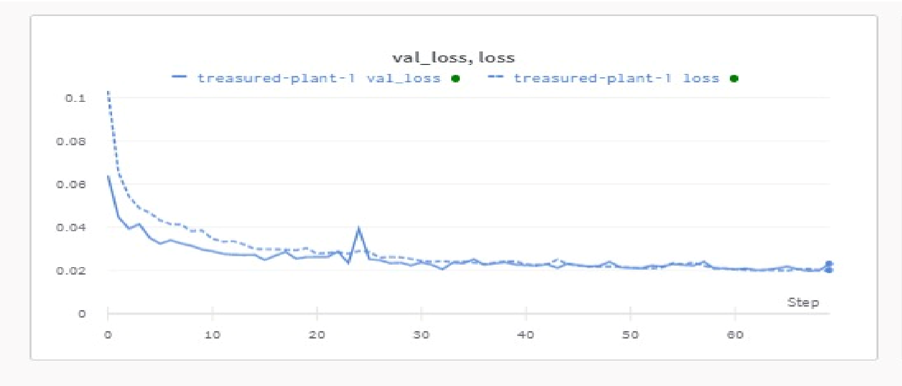
Plot 1: Training and Validation loss (ResNet18)

ResNet50



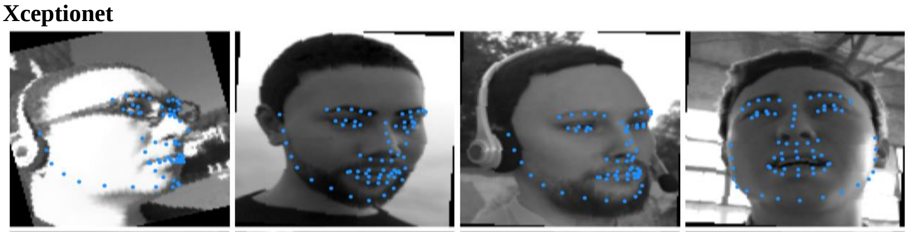
```
Minimum Validation Loss of 0.0147 at epoch 99/100
Model Saved

Epoch(99/100) -> Training Loss: 0.01687667 | Validation Loss: 0.01466818
```

Figure 2: Validation Images and the minimum validation loss (ResNet50)



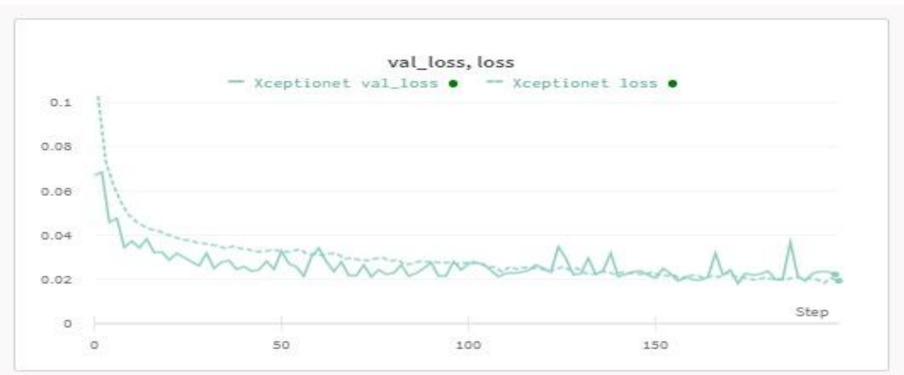
Plot 2: Training and Validation loss (ResNet50)

Xceptionet



```
Saving model...................

Minimum Validation Loss of 0.0184 at epoch 86/100
Model Saved

Epoch(87/100) -> Training Loss: 0.02138573 | Validation Loss: 0.01835696
```

Figure 3: Validation Images and the minimum validation loss (Xceptionet)



Plot 3: Training and Validation loss (Xceptionet)

3. Evaluation

a) 300W Image Dataset

ResNet18

Figure 4: examples of the ResNet18 predictions on 300W images.

ResNet50

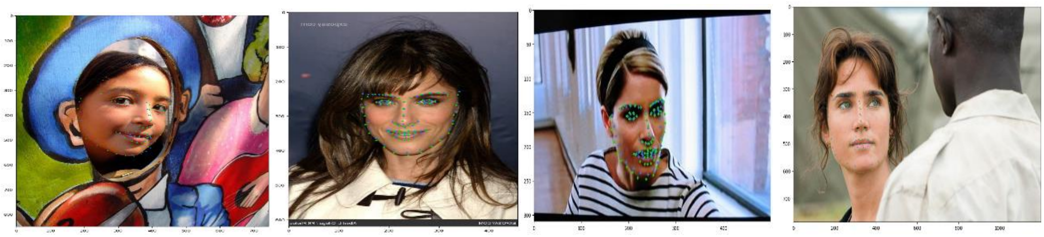


Figure 5: examples of the ResNet50 predictions on 300W images.

Xceptionet

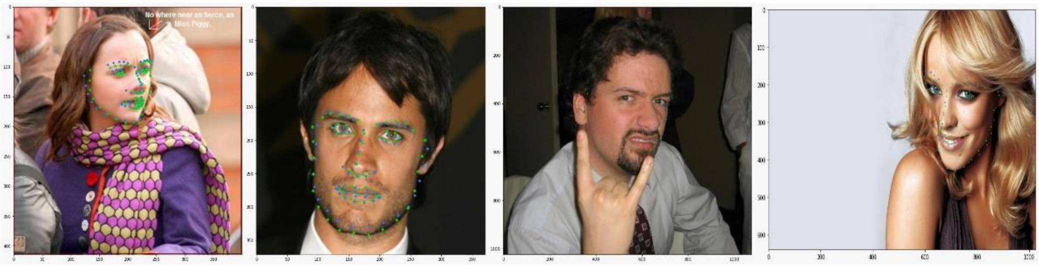


Figure 6: examples of the Xceptionet predictions on 300W images.

b) 300VW Video Dataset and webcam video

The goal is to use the model on live webcam input. To evaluate on actual labelled videos, we use labelled videos from the 300-VW dataset. The videos are all roughly one minute long at about 25-30 frames per second. The dataset has labels for 68 landmarks. For the evaluation we picked four videos (223, 224, 405 and 406) that resemble the situation we are working towards the most that shows the people in front of a webcam or with a similar perspective and distance.

ResNet18 **ResNet50** **Xceptionet**

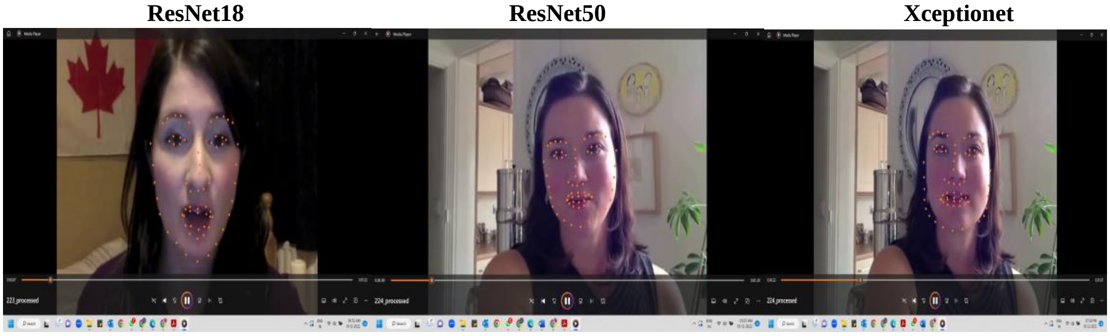


Figure 7: examples of the model predictions on videos(300VW).

4. Discussion

We have used the ResNet18 and ResNet50 models with pretrained weights and Xceptionet model built from scratch to compare the localization performance. As a result, Resnet18 and Resnet50 gave significantly better predictions than Xceptionet on the video dataset. The reliable and valid predictions of the models for both video and image datasets demonstrate that synthetic datasets may be used to train models that will work well with real data. There are constraints. When faces are angled beyond a certain degree and only a small fraction is visible, or the image is too distorted, the model cannot recognize faces adequately. To further improve the accuracy of the prediction we can opt for a subsequent fine tuning of the synthetic-only trained model with different subsets of the real-world datasets.

5. References

1. Wood, E., Baltrusaitis, T., Hewitt, C., Dziadzio, S., Johnson, M., Esterllers, V., Cashman, T., Shotton, J., Fake It Till You Make it, Microsoft - https://microsoft.github.io/FaceSynthetics/
2. Synthetic and Real-World Data in Facial Landmark Detection Models, Datagen,https://datagen.tech/blog/facial-landmark-detection-blog-one/
3. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeirious, S., Pantic, M., 300 Faces In-The-Wild Challenge: database and results, chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_2016_imavis.pdf
4. Rosebrock, A., Facial landmarks with dlib, OpenCV, and Python, https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
5. J.Shen, S.Zafeiriou, G. S. Chrysos, J.Kossaifi, G.Tzimiropoulos, and M. Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In IEEE International Conference on Computer Vision Workshops (ICCVW), 2015. IEEE, 2015.