Team 4: Interim Report

Alejandro Campayo Fernández 7058536
Aishwarya Kshirsagar 7056998
Sai Suresh 7047009

1. Progress Report

1.1. Dataset

1.1.1 First iteration dataset

The dataset used for the first iteration of training consisted of 1000 images from [1], divided into 900 training images and 100 validation images. These images had information on the joints for pose estimation. We used these annotations to mask the arms of the people in the images so that they were suitable for our task. The original images, sized 512x512, were resized to 256x256 in order to decrease the training time needed.

1.1.2 Second iteration dataset

In order to obtain a model with better performance, we gathered data from five open-source datasets: four focused on people ([3], [6], [2] and [5]) and one specific to our task involving Greek sculptures ([4]).

In total, we collected **21,418** images before preprocessing (**1,609** images of Greek sculptures and **19,809** images of people).

We first annotated the joints of the people and sculptures using a pose estimation model from @tensorflow-models/pose-detection. This model identifies keypoints for different body parts and provides:

- **Name:** the body part
- **x:** the x-coordinate of the joint
- **y:** the y-coordinate of the joint
- **Score:** the model's confidence level

We focused on the upper limb keypoints, namely: Right shoulder, Left shoulder, Left elbow, Right elbow, Left wrist, Right wrist.

We processed all images to filter out those where any arm keypoint had a score below 0.4. This helped us remove images that did not contain visible arms.

Next, we created masks either from wrist to elbow (for left or right arms) or from wrist to elbow to shoulder. After this processing, we ended up with **12,094** images:

- **11,629** images of people
- **464** images of Greek sculptures

For each image, we have generate different masked versions by varying the width and height of the mask or by masking different sections of the arm (left/right, lower arm/full arm). All images have been resized to 512x512 pixels and are available in both black-and-white and color versions.

1.2. Model training

Metrics used for Evaluation:
The metrics we are using for evaluating the performance of out models are the following: Dice Coefficient, Pixel Accuracy, SSIM (Structural Similarity Index) and Learned Perceptual Image Patch Similarity (LPIPS), IOU (Intersection Over Union).

For now the loss we are using is MAE on the whole image. However, we are planning on modifying it so that we only compute the loss of the part of the image that has been masked as seen in Fig. 1.

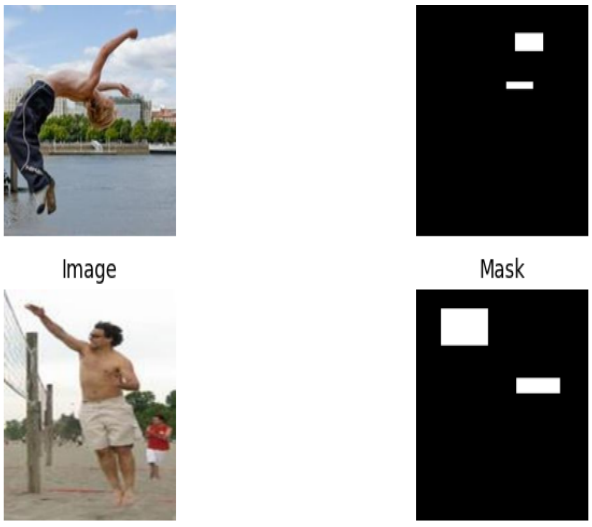

Figure 1

Results after training:
Before training on the Custom Dataset, we tested U-Net model on Cifar-10 dataset. Cifar-10 contains 50,000 train-

ing images and 10,000 test images. The results after training look as follows as in Fig. 2.

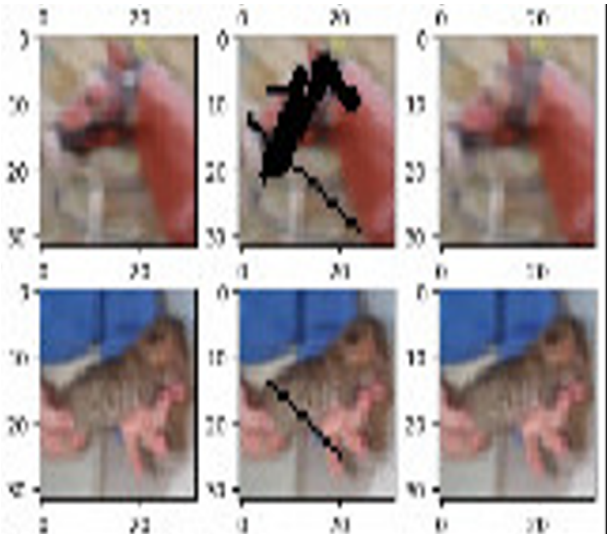

Figure 2

Custom Dataset

After training for 150 epochs on the 900 images of initial iteration we mentioned, we get these results as in Fig. 3.

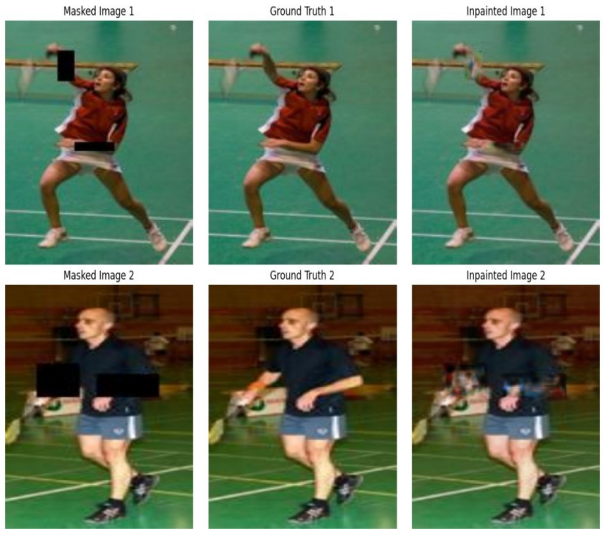

Figure 3

As we see that the results are not as good as the Cifar-10 results.

2. Problems Encountered

2.1. Problems on datasets

We could only use one the datasets mentioned in our project proposal [1] since we found problems when trying to use the rest. Either the data format was invalid or the files to download were too big. This pose estimation dataset used proved to be not sufficient for our task and therefore we had to look for new datasets and annotate them ourselves (Second Iteration Dataset).

2.2. Problems on model training

As we can see above that the results of inpainting on custom dataset is not giving satisfactory results. So, we are planning to train on more data that we have prepared. For training on 25 epochs on colab GPU, we got these results on the evaluation metrics that we used:

Loss: 0.097
Dice Coefficient: 0.507
IoU: 0.340
Pixel Accuracy: 0.003

These results indicate that while the model has learned to some extent, there is significant room for improvement, particularly in pixel accuracy.

Since the results are not good enough, we will be considering training on grayscale images for next iteration. This will not only improve our results but also reduce the training time.

3. Future Work

- We will be including the discriminator model, where adversarial loss is added to the L2 reconstruction loss to get better results.

- When calculating the loss while training, UNet finds the loss of the entire image but we want to find how well the model is able to inpaint the masked region so we will try to create a custom loss function. We have binary mask images where the value is 1 in the region of mask and 0 in the region without mask. By multiplying the Groundtruth and the Predicted image with the binary mask, we set all parts of the image without mask as 0. So while training, the loss is only computed on the regions that consist of the mask as other parts will be set to 0.

References

[1] Sam Johnson and Mark Everingham. Leeds sports pose dataset. https://paperswithcode.com/dataset/lsp, 2010. 1, 2

[2] Niharika. Yoga poses dataset. https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset, 2024. 1

[3] Tapakah68. Segmentation full body mads dataset. https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-mads-dataset, 2024. 1

[4] Thatgeeman. Sculptures of greek olympians dataset, 2024. Accessed: 2024-06-29. 1

[5] TrainingDataPro. Pose estimation dataset. https://www.kaggle.com/datasets/trainingdatapro/pose-estimation, 2024. 1

[6] Papers with Code. Yoga-82 dataset. https://paperswithcode.com/dataset/yoga-82, 2024. 1