Candidate number: 236636
Module: Computer Vision G6032
May 2022

Training a CNN with HOGs to perform face alignment and using the predictions to colour the lips of faces with colour-maps realistically

Introduction

Face alignment outlines the landmarks of a face on an image. The landmarks can then be extracted for modification, such as applying lipstick. We implemented a model that collects the Histogram of Oriented Gradients (HOG) of images of single faces and uses them to train a Convolutional Neural Network (CNN) to predict the face alignment points. We then implemented a script that uses the lip alignment coordinates to realistically change its colour by applying a colour-map. Our HOG-CNN model produces satisfying face alignment results and our lipstick script makes mostly realistic changes to the face but both perform poorly on obstructed faces and dimmer images.

This report will first present our face alignment model, the experimentations that justify its architecture, and its quantitative and qualitative performance against other models. The second part will present our lip-colouring script's method and its qualitative results.

Face alignment with HOG images and a CNN

A. Method overview

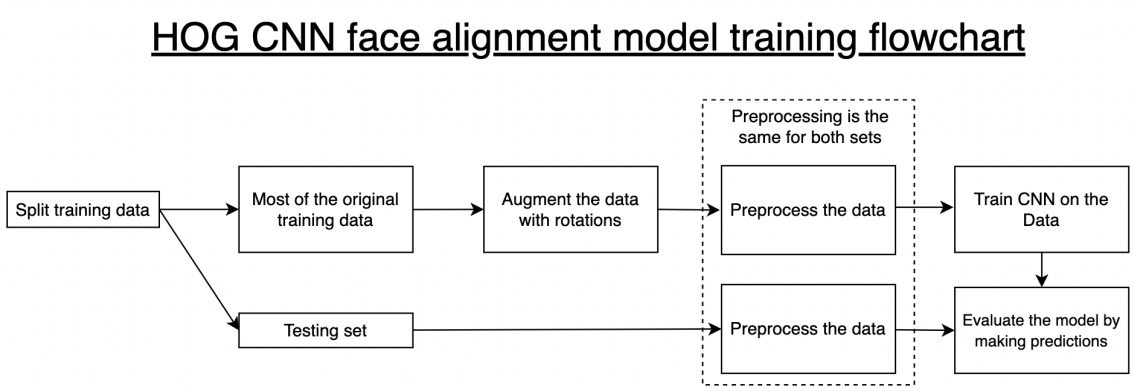


Figure 1. The flowchart of going from our training data to evaluating our face alignment predictions model.

We start by augmenting our training data and processing it to collect the HOGs. We evaluate our model with a part of the training data we did not train our CNN on. Our model is evaluated by the average Euclidean distance between the ground truth points provided in the original training set and the predicted points.

B. Data augmentation

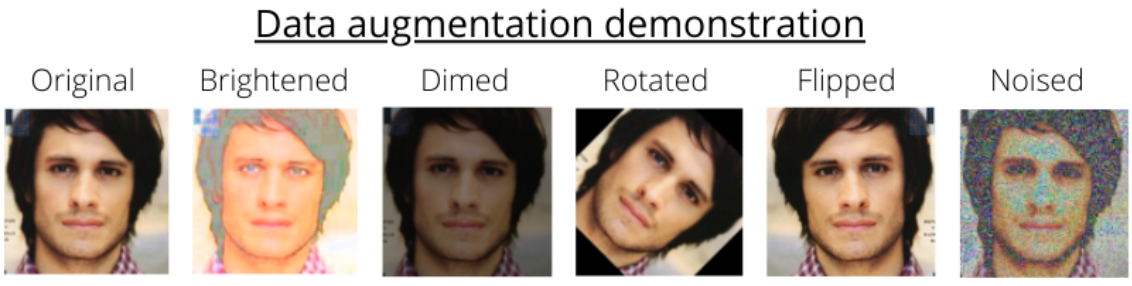


Figure 2. Summary of the effects of possible transformations to augment the training dataset.

We augmented our data by rotating duplicated images but only to realistic extents. We also added random noise to duplicates as we found that it created different HOGs. We excluded some data augmentation techniques (e.g. brightness changes) as they did not change the collected features and including them risked overfitting our CNN. We also excluded flipping the images since the ground truth points are not in a symmetrical order.

Quantitative comparison of our HOG-CNN model trained on augmented data and on non-augmented data

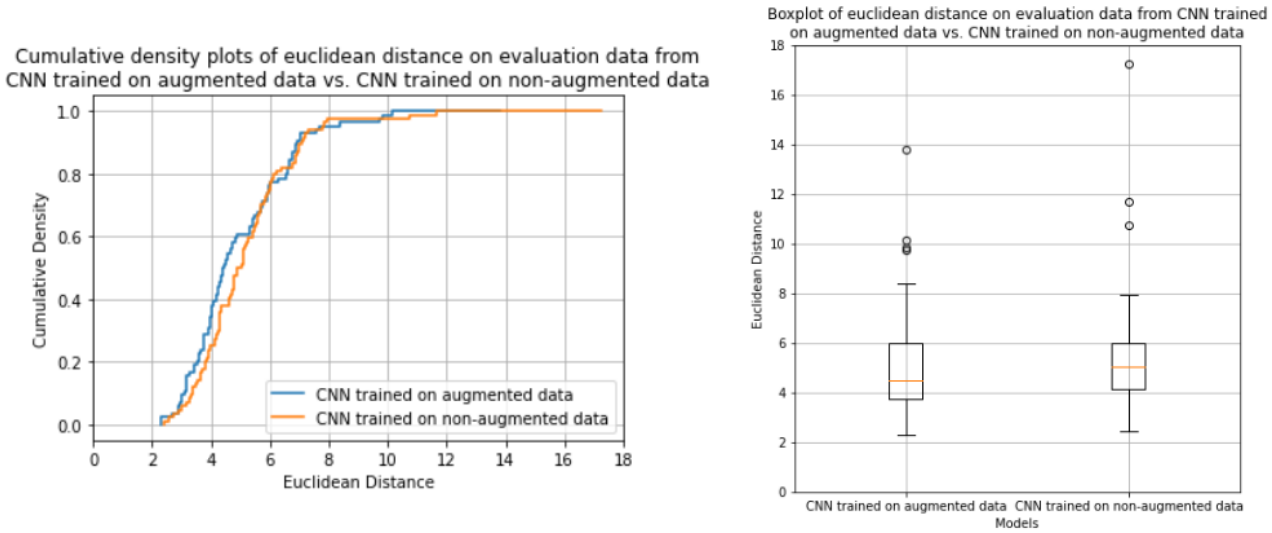


Figure 3. Comparing two models trained on different data: one with the base training data plus the augmented data, and one with only the base training data.

Training our model with augmented data makes it more accurate and lowers its average Euclidean distance, but it does not let it generalise as well as its 1st and 3rd quartiles are further apart than the non-augmented data. Since it has more data to train on, it needs more training time to find generalising patterns.

C. Preprocessing images

The original image compared to its collected HOG features on its 122x122 resized format with different amounts of pixels per cell

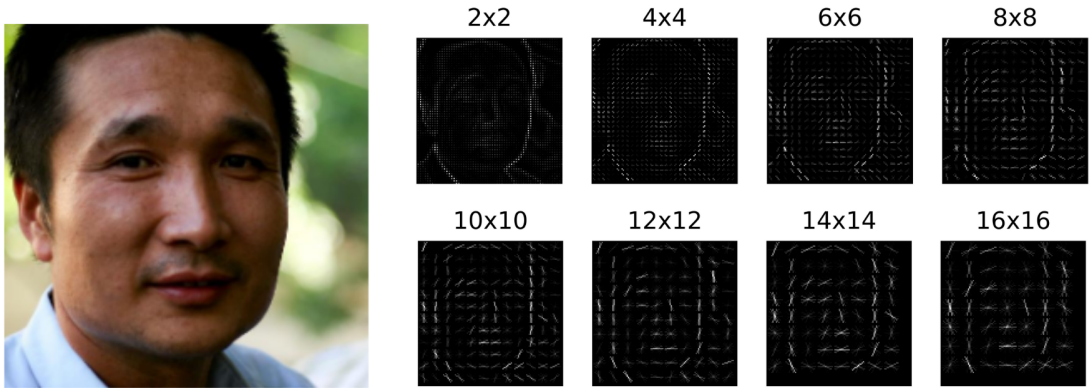


Figure 4. Displaying the HOG features of a 122x122 image with different numbers of pixels per cell.

Quantitative comparison of our HOG-CNN models with different numbers of Pixels Per Cell in the HOG parameters

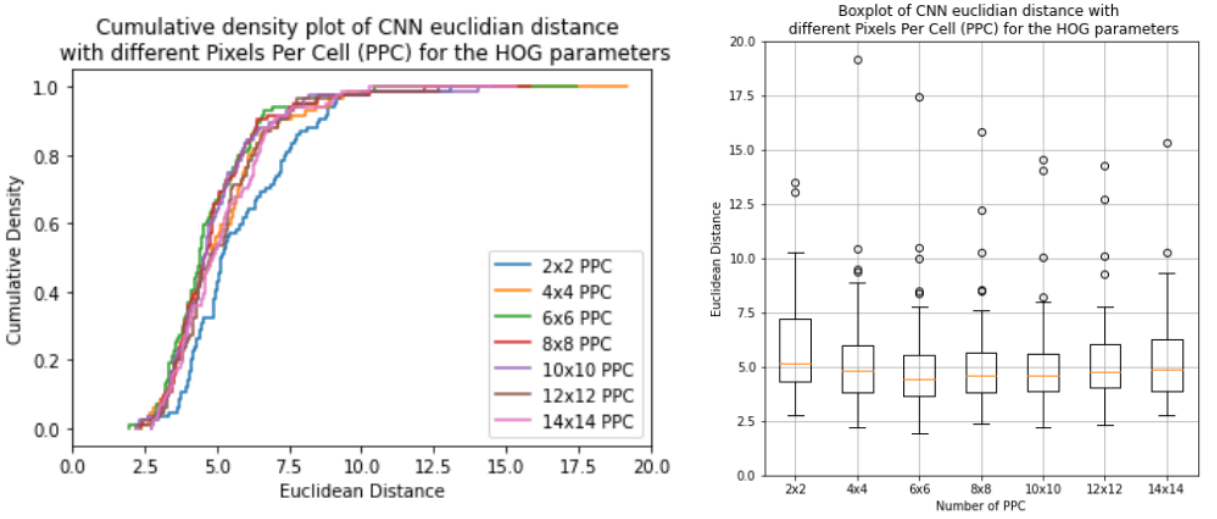


Figure 5. We trained 7 CNNs with training data differing in the preprocessing hyperparameter of the pixels per cell during HOG collection.

We found that HOG features are good at extracting the face from an image and ignoring the rest of the noise (Fig. 4; Singh et al. 2020; Virtanen et al. 2020). Furthermore, being invariant to light changes means changes in lighting should result in the same accuracy from our model's prediction as normal lighting.

One hyperparameter of our HOG feature collector is the number of pixels per cell (PPC). We experimented to find the optimum number of PPC that would result in the most accurate CNN and found it to be 6x6 PPC (Fig. 5).

The images are resized before getting the HOG features which slightly dilates the HOG features compared to the feature collection of the full-size images. Furthermore, the images and points are normalised to standardise our data, improving our CNN's accuracy.

D. <u>CNN model architecture</u>

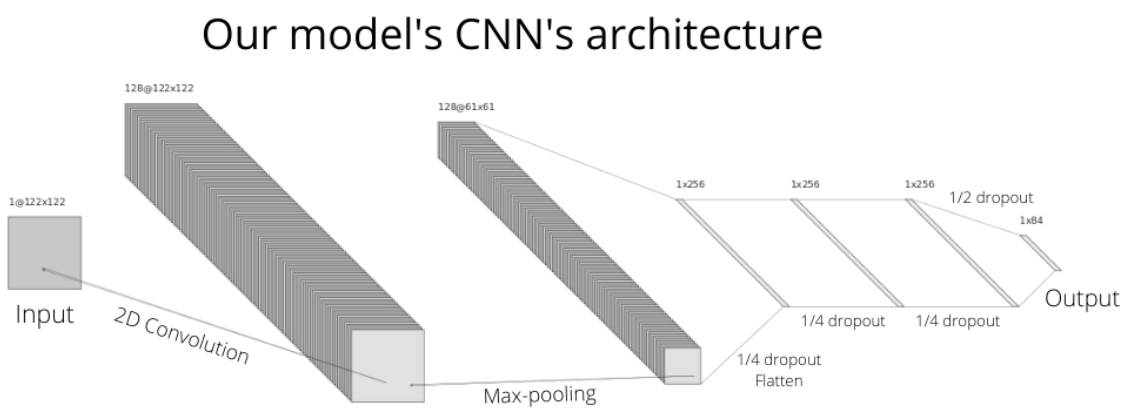


Figure 6. A diagram of our CNN's layers.

We found that the deeper our CNN is, the more accurate it becomes. The first convolutional layer finds spatial patterns in images. The second extracts the most prominent features of convoluted images. They are then flattened into a vector to be passed into 3 dense layers that will find finer details of the image that influence the positions of the coordinates. The dropout at each layer helps prevent overfitting. The size of the final layer is 84, corresponding to the 42 points for facial alignment.

<u>Quantitative comparison of our HOG-CNN model with a Sigmoid activation function on the output layer and ReLU activation function</u>

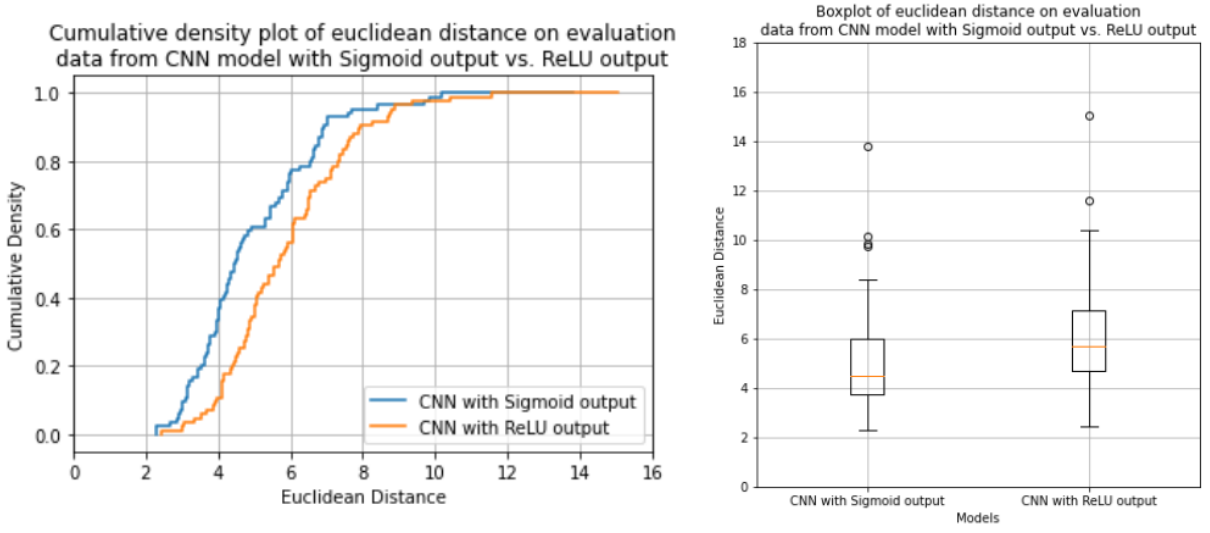


Figure 7. Performance comparison of the same model with a different activation function on the output layer.

Each layer that has an activation function uses ReLU as the numbers we want to predict are non-negative. Only our output layer uses the Sigmoid activation function as we found it improves our model's accuracy (Fig. 7).

E. Prediction model

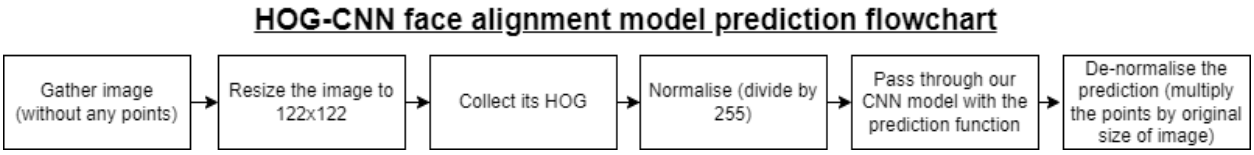


Figure 8. Flowchart to get our model's predicted points on any image (of a face)

This prediction model follows the same steps as our training data with the added de-normalising step to visualise the predictions correctly.

F. Qualitative analysis

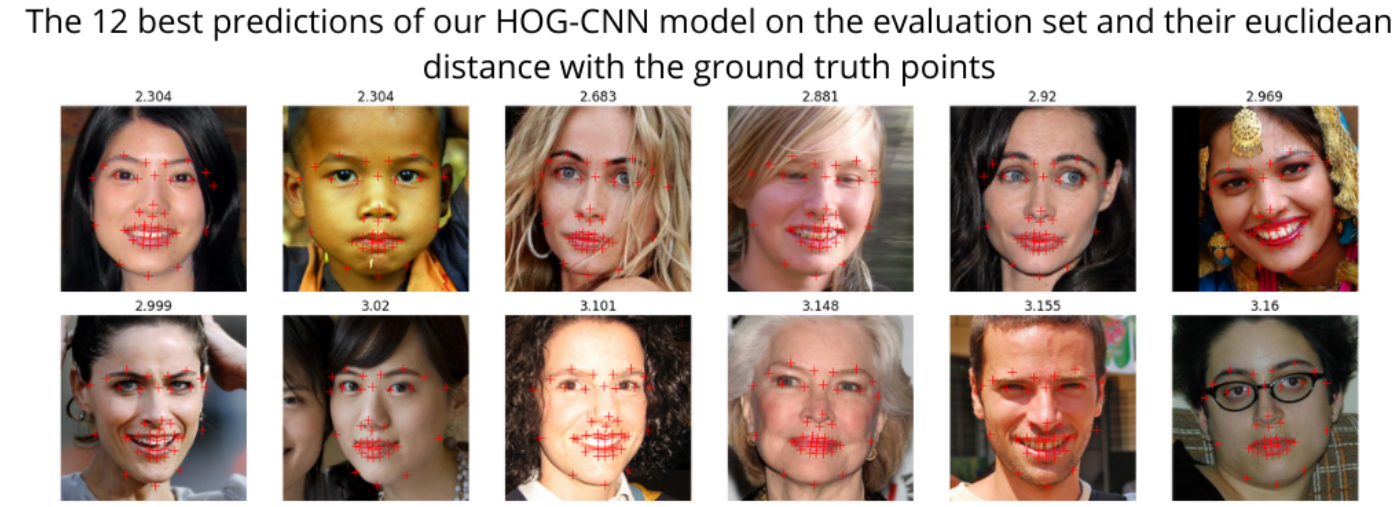


Figure 9. Plotting the 12 best and worst predictions made by our HOG-CNN model on our evaluation set with their Euclidean distance.

Where our model performs well: the best predictions are overwhelmingly on white feminine faces facing straight where there is homogeneous lighting and no obstructions. Our model is good at finding the corners of the eyes and the shape of the face when it is fully in the picture.

Where our model does not perform well: the worst predictions are opposites of most of the best predictions and half of the images are generally darker. Surprisingly, only one image of the 12 worst predictions has parts of its face obstructed. Our model is bad on rotated faces, lack of contrast between face and background and hardly visible lips.

Reasons and solutions: Our training dataset is unbalanced and requires more darker/dimmer images, people with different skin colours than white and better features to extract the lips. We could have made our model more robust using cross-validation. To deal with our model's flaws on faces with non-homogeneous lighting, we could have averaged out the pixel intensity on every image before collecting the HOGs. Our model's flaws in rotated faces show augmenting the data with rotated faces did not make it robust for that and reveals it may be finding general patterns in faces and applying them again.

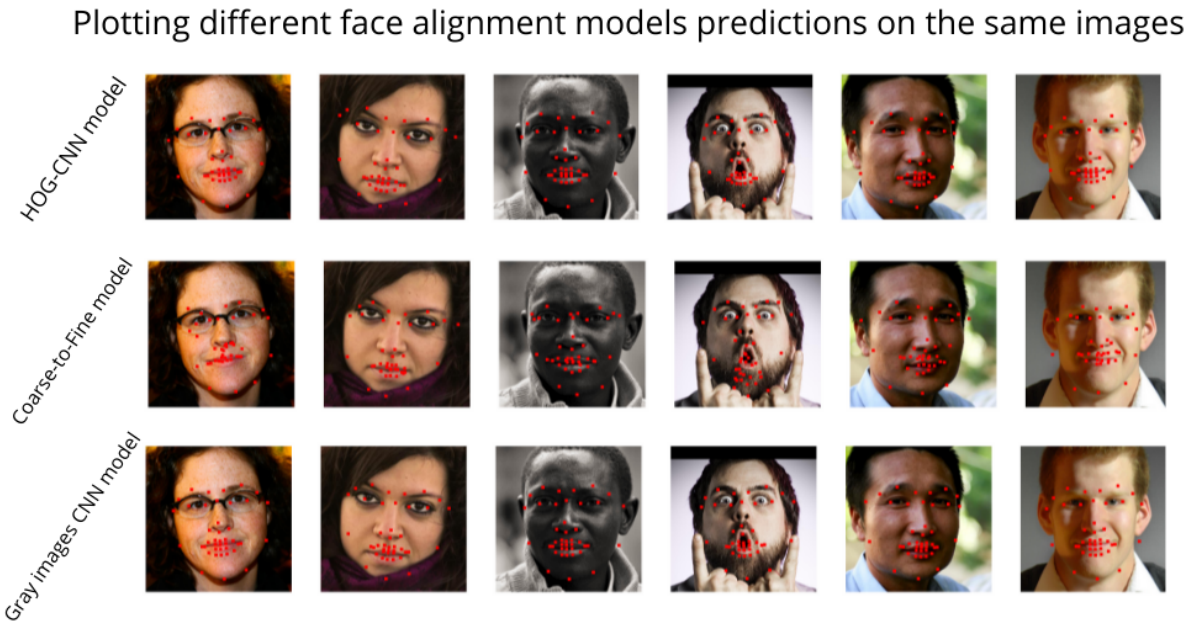


Figure 10. A comparison of the plottings on the same 6 images by different models we implemented.

We have implemented two other models apart from our HOG-CNN model: a simple CNN and a coarse-to-fine regression model (Appendix Fig. 1 & Fig. 2). The coarse-to-fine model seems more robust to shape changes whereas the other two models seem to keep 'standard shapes' for each facial landmark, a characteristic of CNNs, as we can observe with the predicted mouth alignments.

G. Quantitative analysis

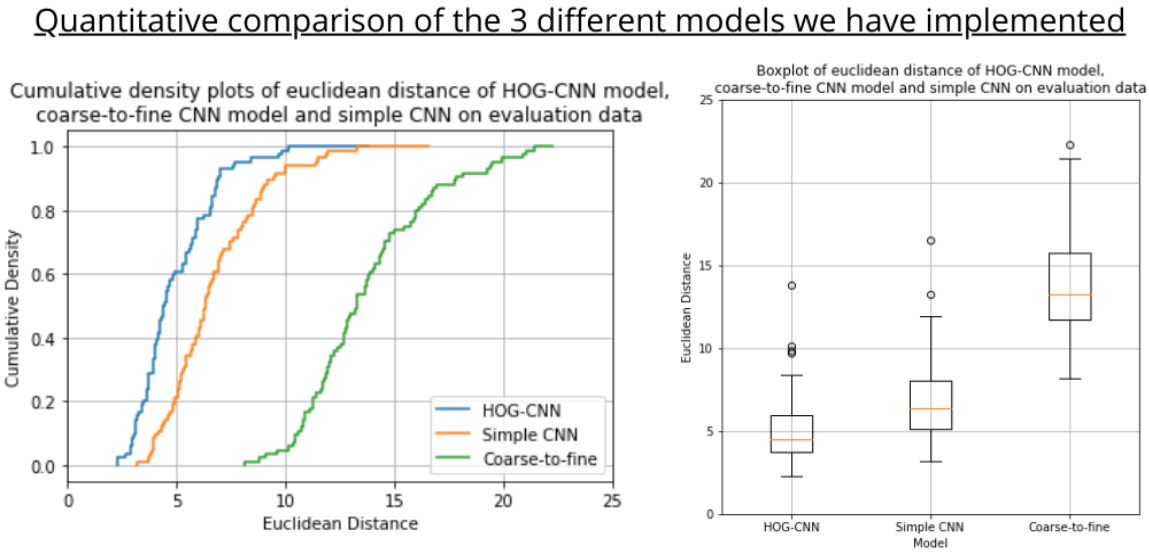


Figure 11. Comparing the performances of 3 different models we have implemented, including the HOG-CNN model we are detailing in this report.

We can see that our HOG-CNN model performs much better than the two other models, getting better euclidean distances overall and a much lower euclidean distance average. Our coarse-to-fine model (Val et al. 2018) seems bad at generalising, as we could observe qualitatively since it deals with more detailed features. Our HOG-CNN model could be improved with an architecture that, similarly to our coarse-to-fine model, takes a closer look at the details of the face that could reveal its alignment, which would improve its robustness to outlier shapes of mouths or face positions.

We can partially attribute the average error of our model to the hand-labelling of the ground-truth. An average difference of 5 pixels between two hand-labellings of the same 244x244 pixels pictures is very reasonable and expected. Therefore, our HOG-CNN model performs, on average, as well as a human.

Face landmark colouring - lips

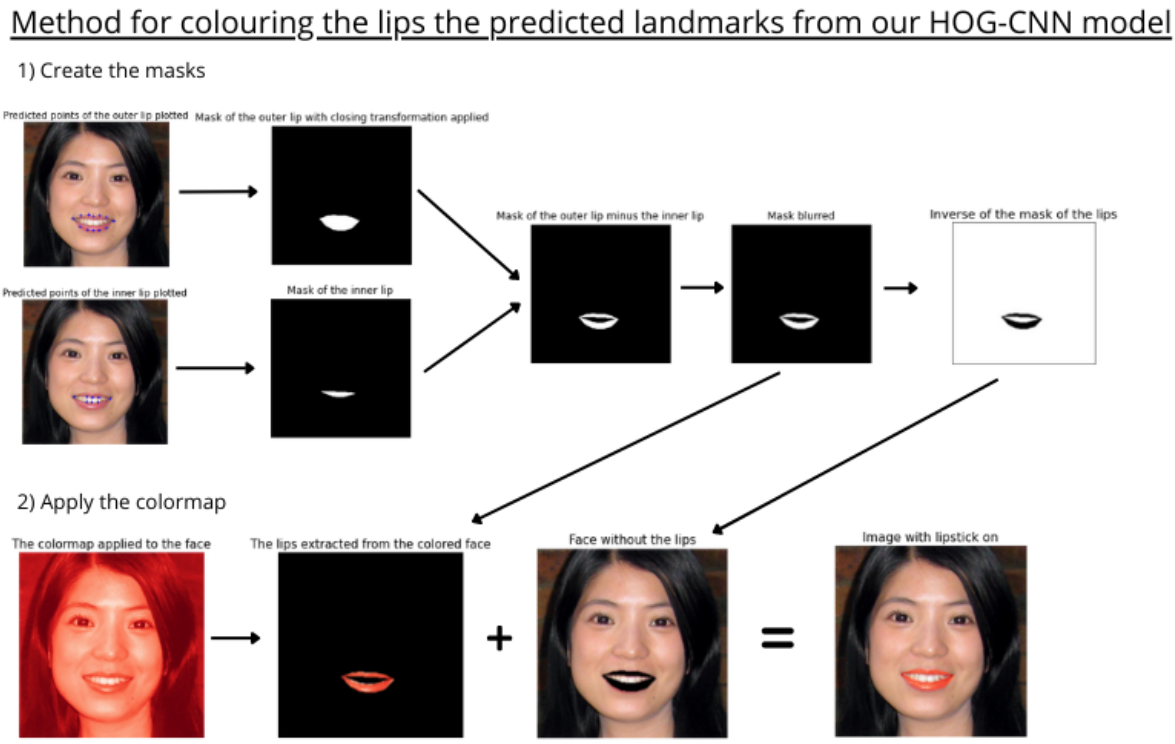


Figure 12. A flowchart of the method we used to colour a face's lips using the predicted points from our HOG-CNN model and colour-mapping.

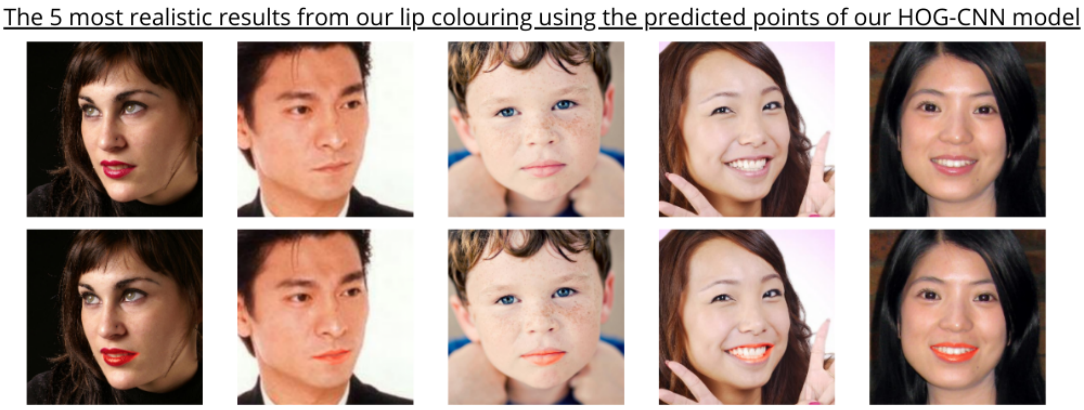


Figure 13. The best qualitative results from our lip colouring method using the HOG-CNN model's predictions to find the lips.

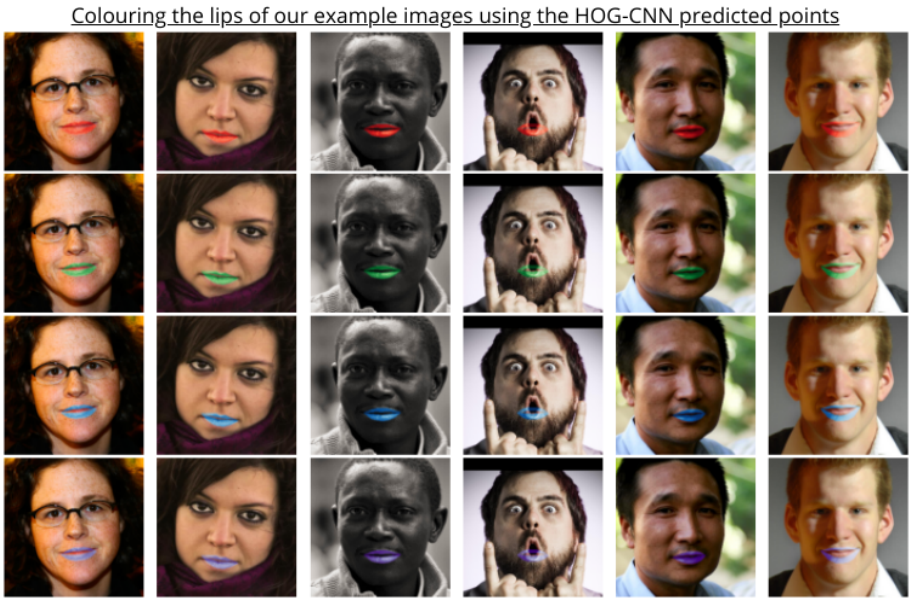


Figure 14. Colouring in the lips of our example images.

We used colour-mapping (Hunter 2007) as it can add a colour filter and not change the texture of the image.

Our results seem to work well on well lit, fully visible lips with an open or closed mouth. Naturally, this is subject to the correct alignment of the lips. Furthermore, less vibrant colours (e.g purple) blend in better with the texture of the lips and look more realistic in darker images.

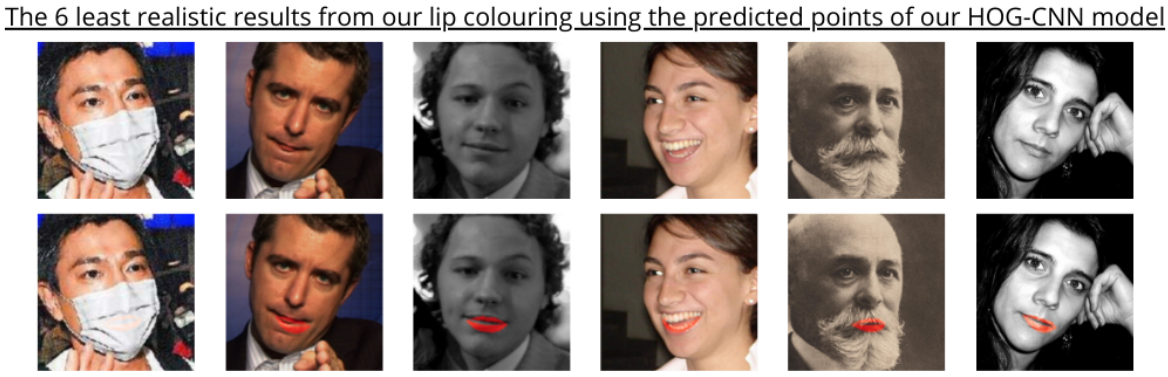


Figure 15. The worst qualitative results from our lip colouring method using the HOG-CNN's predictions.

Our results do not work well on images where the lips are hidden or very thin, but this is also subjective to how well our model has found the lips. Our model also doesn't work on images with a lack of colour (e.g. black and white) or obstructed mouths as the lipstick stands out.

Figure 16. The worst qualitative results from our lip colouring method using the ground truth points.

We can add on the previously listed flaws that our model struggles on oddly shaped lips such as in the 2nd and 3rd images.

Conclusion

In conclusion, we have implemented a face alignment prediction model that uses HOGs of images to train a CNN that performs very well on well-lit images and moderately on darker and obstructed ones. We affirm that with the expected divergence in labelling, our model performs, on average, as well as a human labeller. Furthermore, we have implemented a realistic virtual lipstick applier using the predicted landmarks from our HOG-CNN model and colour-maps. Our models could still be improved through data balancing, cross-validation, and histogram averaging to get better performances from our CNN and our lipstick applier.

References

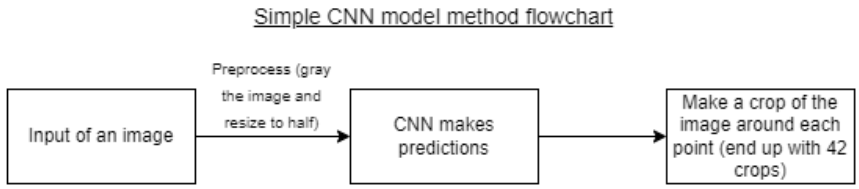
Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

Singh, Swarnima & Singh, Durgesh & Yadav, Vikash. (2020). Face Recognition Using HOG Feature Extraction and SVM Classifier. 8. 6437-6440. 10.30534/ijeter/2020/244892020.

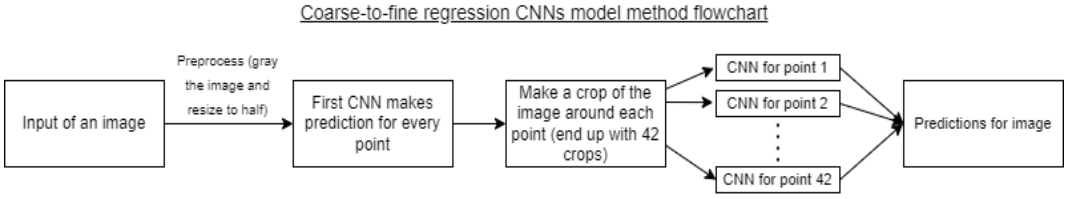
Valle, R., Buenaposada, J. M., Valdés, A., & Baumela, L. (2018). A Deeply-Initialized Coarse-to-fine Ensemble of Regression Trees for Face Alignment. ECCV (14), 609–624.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 261–272.

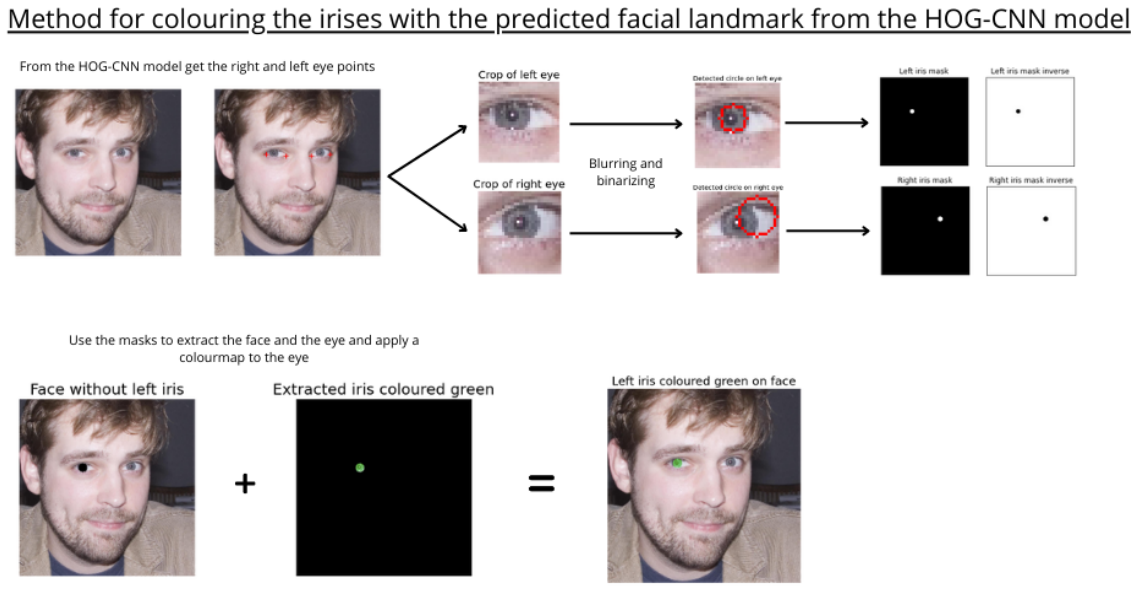
Appendix



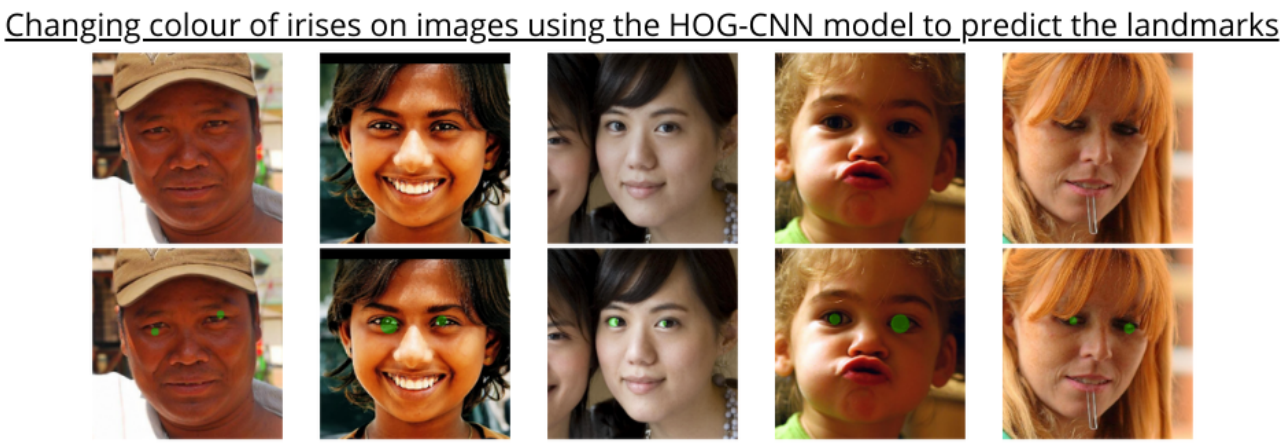
Appendix Figure 1. Flowchart of our simple CNN model, one of the models we implemented and used to compare the performance with our HOG-CNN model. The images go through very basic preprocessing and do not collect HOG features.



Appendix Figure 2. Flowchart of our coarse-to-fine model we implemented and used to compare the performance with our HOG-CNN model. The images go through very basic preprocessing and do not collect HOG features. This model trains 43 models, the first takes as input the 122x122 image and predicts all of the points and we then use these predictions and make a crop around each to train 42 models that are each specialised in training one specific point on the image (e.g. the tip of the nose). We then concatenate these predictions to return them as the face alignment predictions for the image.



Appendix Figure 3. A flowchart of the method we used to colour in the iris of a face using the predicted points from our HOG-CNN model and colour-mapping.



Appendix Figure 4. Here are images with the attempts to change the colour of the irises. Our method fails for two reasons: the first is that it does not always find the contour of the iris, making the next steps in the method impossible, and if it does return a contour it's not always exactly the iris. The iris needs to be fully visible (e.g. right eye of example image 3 in the figure) for our method to find it. We can improve our method by finding a more robust function to find the circle of the irises or process the image differently at a higher resolution. The results of this method are unrealistic.