Computer Vision: Face Alignment and Lips/Eye Colour Modification Assignment

Anthony Guerges - CandNo 262809

May 2024

1 Abstract

Face alignment is crucial for various computer vision applications, relying on accurate detection of facial landmarks. It is vital for applications in facial recognition, emotion detection, and augmented reality. In this report, we look into tackling the task of training a model to predict landmarks on the given images. We portray that using feature extraction and regression leads to a dependable and successful model.

2 Introduction

This study aimed to develop a reliable and robust face alignment system by testing various methods, including Convolutional Neural Networks (CNN) and regression models. The development process included preprocessing, image augmentation, and feature extraction using Scale-Invariant Feature Transform (SIFT). Our analysis shows that the Random Forest (RF) model achieved superior results, demonstrated by cumulative error distribution (CED) and error box plots. This report outlines our approach and findings in improving the model's accuracy and generalizability.

3 Methodology

3.1 Preprocessing

The preprocessing steps include resizing images, converting them to grayscale [2], and normalizing pixel values. This ensures that the images are standardized for further processing and feature extraction.

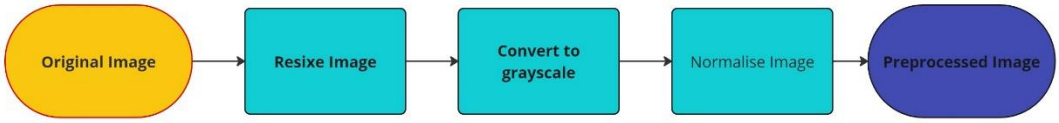

Figure 1 Preprocessing Flowchart

Each image is resized to a target size (256x256 pixels), making the dataset uniform and reducing computational complexity. Then converted to grayscale using OpenCV's cv2.cvtColor function, Simplifying the image data, focusing on intensity values and reducing the dimensionality. The grayscale image is then normalized by dividing pixel values by 255.0 to scale them between 0 and 1. Grayscale normalization reduces computational complexity by focusing on intensity variations and removing colour information, which is often redundant for feature extraction in many vision tasks.

3.2 Image Augmentation

Image Augmentation: Techniques like rotation, shift, brightness adjustment, shear, zoom, and horizontal flip are employed to artificially increase the dataset size and variability. This helps in making the model robust to variations in the face images, thus improving generalization [5]. I also explain how I've used data augmentation in improving singular deficiencies in the models performance in the qualitative analysis section of the report.

```python
def get_data_generator():
    return ImageDataGenerator(
        rotation_range=30,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest'
    )
```

Figure 3 Augmentation Code

Figure 2 Data Augmentation Example [6]

3.3 Feature Extraction

We use the Scale-Invariant Feature Transform (SIFT) to extract features from images. This step involves detecting keypoints [4] and computing descriptors,

followed by padding or truncating the descriptors to a fixed size.

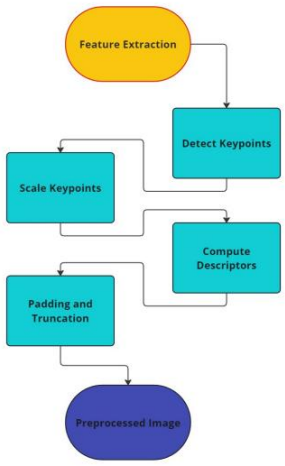

Figure 4 Feature Extraction Flowchart

SIFT is used to detect keypoints and compute descriptors. Each image is converted to 8-bit, required by OpenCV's SIFT implementation. keypoints are first detected, then scaled by 'scale factor' of 1.6 and a 'number of scales' of 5. A scale factor of 1.6 means that each octave (a group of scales) is divided into layers, with each layer being 1.6 times larger than the previous one. Setting the number of scales to 5 means that each octave in the scale space pyramid contains five different scales at which keypoints are detected. The choice of these values is so that the feature extraction process is robust and efficient; it means that the algorithm can handle variations in image size and resolution, as well as capturing essential features.
The descriptors are then computed for the detected keypoints - SIFT descriptors capture invariant features that are crucial for identifying key points in the images. These descriptors are then padded/truncated to a max of 500 features. Limiting the number of features to 500 helps in managing the dimensionality of the data, making the training process more efficient while retaining sufficient information from the images. The resulting feature vectors are flattened in order to create a fixed-size feature vector.

3.4 Prediction Model

We employ a Random Forest Regressor [1]as our prediction model. The dataset is split into training and validation sets, and Principal Component Analysis (PCA) is used to reduce dimensionality before training.

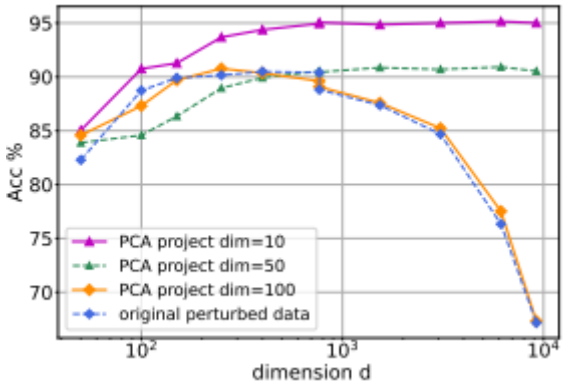

Figure 5 Example of PCA improving robustness of a model [6]

Random Forest Regressor: A robust, ensemble-based model that reduces overfitting and improves accuracy by averaging multiple decision trees. It works by using Bootstrap Sampling, where multiple subsets of the original training data are created using random sampling with replacement, constructing a tree for each sample. Using 100 trees in the forest ensures that the model has enough diversity to make accurate predictions. I have used regularisation in multiple instances, such as:
-Max Depth: Limits the maximum depth of each decision tree to prevent overfitting by restricting the complexity of the model.
-Min Samples Split: This ensures that the model does not create overly specific splits, which helps in maintaining generalization. I selected the value 10 so that a sufficient amount of samples are chosen to split an internal node.
-Min Samples Leaf: Sets the minimum number of samples that a leaf node must have, which ensures that leaf nodes have a sufficient number of samples. I set a minimum of 4 samples per leaf node to prevent the model from creating leaf nodes with very few samples, which would lead to overfitting.
PCA [3] is used for dimensionality reduction, applied to SIFT features to reduce dimensions from 500x128 to 100 principal components. This balance retains significant information while reducing noise, speeding up training and improving model generalization.

4 Results and Discussion

4.1 Quantitative Analysis

Convolutional Neural Network (CNN) My initial model was a Convolutional Neural Network (CNN) designed with several convolutional layers, batch normalization, max-pooling layers, and fully connected layers. The model's architecture included several features such as:
Pooling Layers: Reduce spatial dimensions while retaining essential features. Fully Connected Layers: Integrate learned features for landmark prediction. Dropout Layers: Prevent overfitting by randomly dropping neurons

during training. To evaluate and compare the performance of the CNN model and the Random Forest model, I used a Cumulative Error Distribution (CED) and included a box plot to compare error values.

Discussion: After looking at both metrics, it was clear to see that the Random Forest model was more accurate, indicating better accuracy and consistency in predicting facial landmarks.

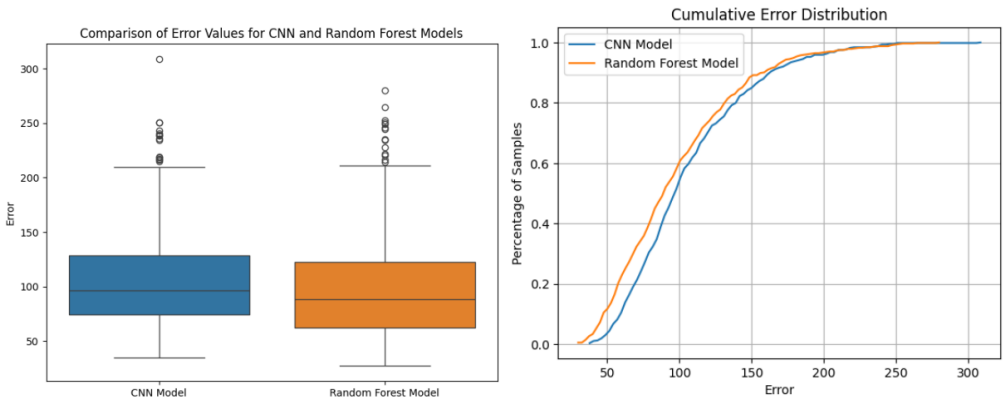

Figure 6 Box Plot and CED comparing both models

4.2 Qualitative Analysis

After selecting the random forest model over the CNN model, I thought to analyse where it struggled most with predicting landmarks. One of the main struggles the model had was adjusting the landmarks when the face was rotated to either side. The model also seemed to predict landmarks with higher error when performing on different skin colours. In order to mitigate this I made some adjustments to the model.

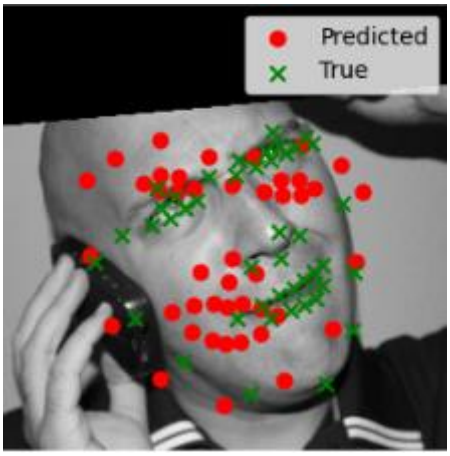

Figure 7 Example of predicted landmarks working badly with tilted heads

Addressing tilted heads was the main reason behind including data augmentation (including head rotation) in my model. Although data augmentation alone may have improved the model performance on varying skin colours, I decided to implement a more sophisticated model architecture. I included a PCA in order to standardize features across different faces. My idea was that a PCA would potentially reduce some inherent biases by focusing on the most significant features that vary across all faces, regardless of ethnicity.

Figure 8 Improved model use on Example Images

5 Face Alignment Conclusion

This report presents a detailed overview of the development and evaluation of a face alignment system. For future work, looking into combining regressors such as SIFT and Histograms of Oriented Gradients (HOG) could be an interesting matter.

6 Eye/Lip Modifier

6.1 Overview

The goal of this segment of the assignment was to design and implement a system for modifying the colour of lips and eyes in an image. I decided to challenge myself by attempting a solution without using my predicted landmarks. My solution employs an algorithmic procedure based on colour segmentation using HSV (Hue, Saturation, Value) colour space and morphological operations to achieve the desired modification.

6.2 Implementation Steps

The steps I took in making this modifier are outlined in the figure below.

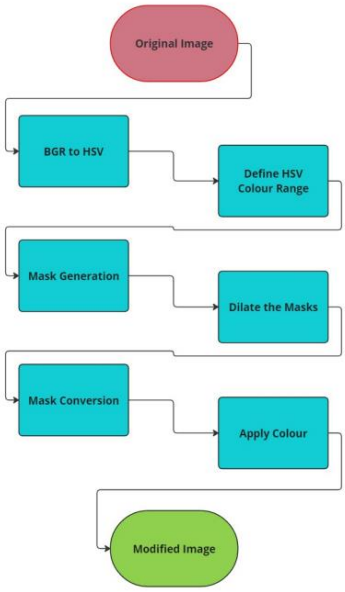

Figure 9 Eye/Lip modifying flowchart

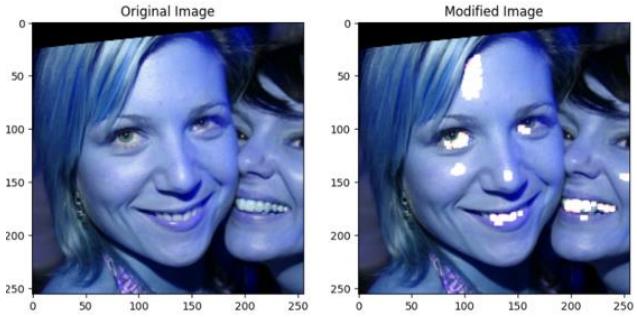

Figure 10 Example usage of Lip/Eye modifier

• Colour Range Definition: Define HSV colour ranges to isolate lips and eyes. For lips, the range focuses on red hues, while for eyes, it captures very light regions which typically indicate the whites or light-coloured eyes.

• Mask Generation: Create binary masks for the lips and eyes using the defined colour ranges with cv2.inRange(). This isolates the regions of interest.

• Morphological Cleaning: Dilate the masks using a 5x5 kernel to close gaps and noise in the segmented regions.

• Mask Conversion: Convert the binary masks to 3-channel format to facilitate colour application.

• Colour Application: Create images filled with the target colours for lips and eyes. Apply these colours to the original image by using bitwise operations with the masks.

• Combine Modified Regions: The coloured regions are added back to the original image to produce the final modified image.
Here are some examples of the Model being applied. One downside of using this type of model is that teeth are hard to distinguish from the eyes as they are a similar colour.

References

[1] Leo Breiman. Random forests. Machine learning, 45:5–32, 2001.

[2] Rafael C Gonzalez. Digital image processing. Pearson education india, 2009.

[3] Ian T Jolliffe. Principal component analysis. Technometrics, 45(3):276, 2003.

[4] David G Lowe. Distinctive image features from scale-invariant keypoints.

International journal of computer vision, 60:91–110, 2004.

[5] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. Journal of big data, 6(1):1–48, 2019.

[6] Xiang Wang, Kai Wang, and Shiguo Lian. A survey on face data augmentation for the training of deep neural networks. Neural Computing and

Applications, 32(19):15503–15531, March 2020.