# Task 1: CNN Implementation*

*Comparative analysis of CNN with hog and sift

Areeba Fatah (21i-0349)
*AI (artificial intelligence )*
*FAST, NUCES)*
Islamabad, Pakistan
i210349@nu.edu.pk

## I. Introduction

This tasks involves segmenting the signatures from grids stored in an image, extracting features using CNN and classifying the signatures.More we do comparative analysis with Hog and Sift feature extraction. The comparative analysis of these involves visual representations and evaluation metrics like accuracy score, f1 score.

## II. Methodology

The data we got was of 16 images. Each image had a grid where each row belonged to a person having an id and 4 instances of signatures. Each image had 12 rows so total individuals are 192 3. . The task further involves the comparative analysis of CNN with hog and sift. The following the elaborated detail of methodologies used:

### *A. Preprocessing and Segmentation of signatures*

- **Division of image into rows** Since each image had 12 rows so it was important to divide the image so the first step was to divide the image into 12 parts.The image is not aligned well and the signature are out of the box too. To do this, I got estimated height, width of the signature box as well as the Id box using the image editor as well as the code. The approximate dimensions are height =525 , width of signature box =1193 and id box = 392. After this the division rows was relatively easy. Result in 17.
- **Division of one row into signatures** After getting one row we can further segment it into parts that is signatures. So here I'll use the above mentioned dimensions to get one signature then preprocess it by removing lines (which could lead to pseudo signature features this is done by using cv2 morph function ), removing noise generated by lines removal with gaussian then binarizing the image , dillating it for prominent and continuous lines and getting contours. For ease of reference and need i have save both the preprocessed signatures and the segmented ones. Refer to 2 and 1.
- Repeating it all for all the images in the directory.

### *B. Training with CNN*

- **CNN architecture** The CNN model was based on keras. it had two conv layers each followed by max pooling layer to reduce the dimensions and the classification part had one flattened layer then one dense layer followed by dropout layer and softmax.
- **Grid Search for optimal parameters of CNN layers and classifier layers** Here i trained the CNN on multiple sets of parameters i.e filters, dropout, learning rate and epochs. In this part multiple sets of hyperparameters were provided to see which ones are the best.The training details are provided in the next lines followed by this point. The optimal parameters were Best results = Filters: 64, Dropout Rate: 0.3, Epochs: 10, Learning Rate: 0.0001.
- **CNN training on the best parameters** For training a data loading function was created whose purpose was to load the images and their corresponding labels. Then i normalized the images for better results to get rid

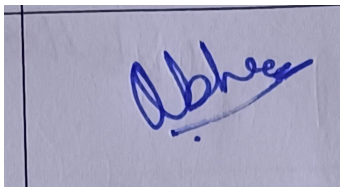

Fig. 1. Unpreprocessed image

Fig. 2. Preprocessed image

Fig. 3. A glimpse of dataset

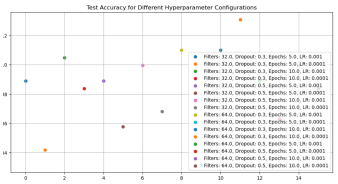


Fig. 4. Grid Search Results for CNN

of outliers , reshaped the images to match the model's dimensions. Just for my better understanding i separately CNN on both processed and unprocessed images. For evaluation i used accuracy, recall , f1 score, classification report , confusion matrix (which was not suitable here) and visualized the results which will be discussed in results section.

### C. Training ANN with HOG and SIFT

- **ANN architecture** The architecture had one dense layer followed dropout similar to CNN then another dense layer with ReLU activation and then layer with softmax.
- **HOG feature extraction** The only change here in comparison to CNN is ANN use without conv layers and an additional function to get HOG features from the images. Here again i used both unpreprocessed and processed images.
- **SIFT feature extraction** Here i have used sift with 128 keypoints and the rest is the same as HOG.

## III. Results

The results of preprocessing were already discussed above so here we'll be discussing results of training i.e. training with CNN and ANN with HOG and SIFT feature extraction techniques on preprocessed and unpreprocessed signature image.

### A. CNN Results

- **Grid Search Results** Given list of hyperparameters
  - filters=[32,64]
  - dropouts=[0.3, 0.5]
  - epochs=[5, 10]
  - learning rates=[0.001, 0.0001] The best results were at Best results = Filters: 64, Dropout Rate: 0.3, Epochs: 10, Learning Rate: 0.0001, Test Accuracy: 13.09% . Visualized at 4.
- **CNN training on preprocessed images** By preprocessed i mean images like 2. i.e. dilated, noise free , line free images. The results are
  - Test accuracy: 10.47%
  - Training accuracy: 94%
  - Precision: 11.74%
  - Recall: 10.47%
  - F1-Score: 9.83%

  Visualizations of both loss and accuracy of validation and training can be seen at 5 and 6.
- **CNN training on unpreprocessed images** By unpreprocessed i mean images like 1. which is just a part of the dataset image with nothing done on it except division. The results are
  - Test accuracy: 10.99%
  - Training accuracy: 90%
  - Precision: 11.40%
  - Recall: 10.99%
  - F1-Score: 10.99%

  Visualizations of both loss and accuracy of validation and training can be seen at 8 and 7 .

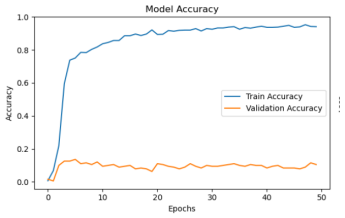


Fig. 5. Accuracy of CNN on Preprocessed Images

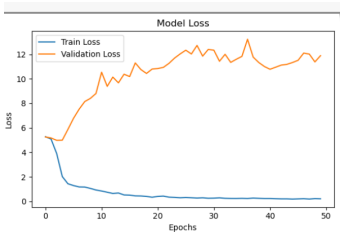


Fig. 6. Loss of CNN on Preprocessed Images

### B. ANN training on HOG features

- **On Preprocessed Images** the results are
  - Test accuracy: 21.47%
  - Training accuracy: 94%
  - Precision: 24%
  - Recall: 21%

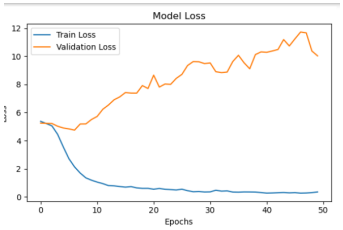


Fig. 7. Loss of CNN on unpreprocessed data

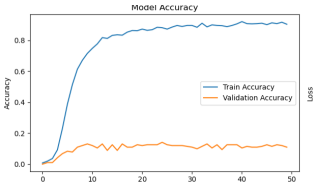


Fig. 8. Accuracy of CNN on unpreprocessed data

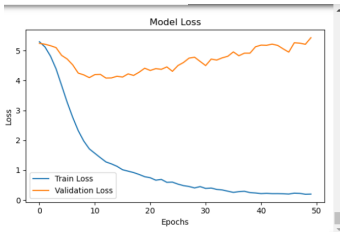


Fig. 9. Loss of ANN with HOG features

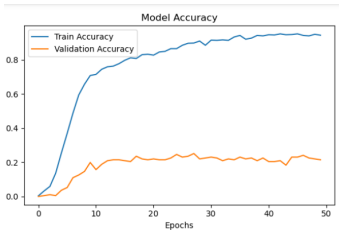


Fig. 10. Accuracy of ANN with HOG features

  - F1-Score: 20%

  Visualizations of both loss and accuracy of validation and training can be seen at 10 and 9 .

- On Unpreprocessed Images the results are
  - Test accuracy: 4.71%
  - Training accuracy: 17%
  - Precision: 0.5%
  - Recall: 0.5%
  - F1-Score: 0.4%

  Visualizations of both loss and accuracy of validation and training can be seen at 11 and 12 .

### *C. ANN training on SIFT features*

Accuracy: 1.57Precision: 0.00 Recall: 0.02 F1 Score: 0.01

- **On preprocessed data** the results are

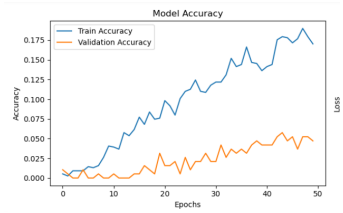


Fig. 11. Accuracy of ANN with HOG on unpreprocessed data

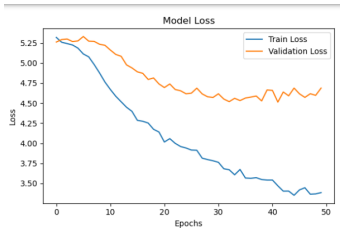


Fig. 12. Loss of ANN with HOG on unpreprocessed data

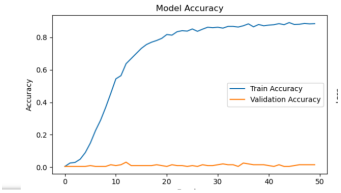


Fig. 13. Accuracy of ANN with SIFT on preprocessed images

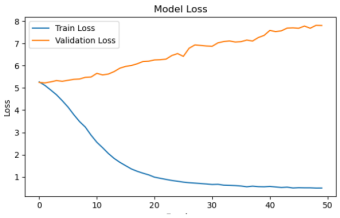


Fig. 14. Loss of ANN with SIFT on preprocessed images

  - Test accuracy: 1.57%
  - Training accuracy: 88%
  - Precision: 0%
  - Recall: 2%
  - F1-Score: 1%

  Visualizations of both loss and accuracy of validation and training can be seen at 13 and 14 .

- **Unpreprocessed data** the images need to be in grayscale though so the results are
  - Test accuracy: 8.90%
  - Training accuracy: 42%
  - Precision: 10%
  - Recall: 9%
  - F1-Score: 8%

  Visualizations of both loss and accuracy of validation and training can be seen at 15 and 16 .

### *D. Findings*

- CNN was obviously over-fitting with very little generalization in the preprocessed images. i.e high variance. However its accuracy was slightly better in unpreprocessed images i.e. slightly low variance but f1 score , recall and precision says otherwise.
- Surprisingly HOG performed better having test accuracy of 21% . This could be because of preprocessing steps

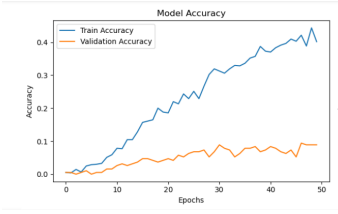


Fig. 15. Accuracy of ANN with SIFT on Unpreprocessed images

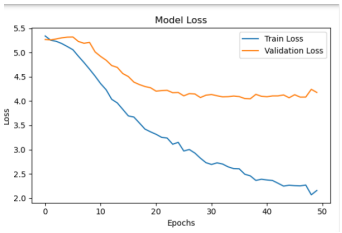


Fig. 16. Loss of ANN with SIFT on Unpreprocessed images

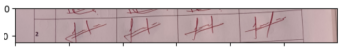

Fig. 17. One row after division

as in the unpreprocessed images its performance deteriorated.Hence preprocessing is important for HOG.

- Sift didn't perform well on preprocessed images since it couldn't have found any good key descriptors in black and white images. It's performance increased a bit in the other images but it was not very good. I

## IV. Discussion

- The first reason of not so good performance could be lack of large training data that could help avoid over-fitting.
- Another reason could be the alignment of images was not perfect hence we couldn't form very good bounding boxes and segment the images. Also the images had grid lines which induced pseudo features in the data leading to model's confusion.
- Another reason could be that the data is simple without much variety and samples such that CNN due to its complexity over fits it.
- Sift can't work well without key descriptors like shown in its performance on preprocessed images.
- Also seeing HOG's better performance along with CNN we can say that preprocessing had a good impact on the performance.

## Conclusion

CNN which is well known for its performance in classification tasks ,surprisingly didn't yield any strong results due to over fitting. It could also be due to not very good segmented data. This maybe due to small dataset or task being too simple in front of CNN. For this task HOG features with ANN were most suitable after preprocessing the image. SIFT performance was the worst maybe because the data lacks too many descriptors. To conclude it all, having a good model like CNN doesn't mean good accuracy, it also depends on the data.

## Prompts

- "Give me code for getting signatures from images containing grids, where each row belongs to a person and has 5 parts id and 4 columns for signatures."

TABLE I
Model Performance Metrics

| Model | Preprocessing | Test Accuracy (%) | Training Accuracy (%) |
|---|---|---|---|
| CNN | Preprocessed | 10.47 | 94.00 |
| CNN | Unpreprocessed | 10.99 | 90.00 |
| ANN (HOG) | Preprocessed | 21.47 | 94.00 |
| ANN (HOG) | Unpreprocessed | 4.71 | 17.00 |
| ANN (SIFT) | Preprocessed | 1.57 | 88.00 |
| ANN (SIFT) | Unpreprocessed | 8.90 | 42.00 |

- " Read all the images in a folder and pass through signature extractor function."
- "Take an image with grid and remove lines, noise and dilate it."
- "Give me code for optimizing CNN hyperparameters like filters, dropout rate, and learning rate using grid search."
- "Give me an code of how to add calculate evaluation metrics (accuracy, recall, F1-score) for training a CNN on the given images."
- "Give me code for extracting features using HOG and SIFT for signature recognition, and compare their performance to CNN.Complete code like CNN."
- "Give me code to visualize and interpret the loss and accuracy of CNN and ANN models during training as i have given you."
- "Analyze these and tell me why CNN is not working well."
- "Convert all these findings to table in latex"

## References

G. Eason, "handwritten signature detection python," YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=cBtyoF9UYA0. [Accessed: 2024-09-27].