OCR text extraction in medical images

Akashdeep Singh

University of Calgary

INTRODUCTION

With over millions of patient's medical scans performed, significant and sensitive data needs to be protected from the working staff, without compromising the evaluations of the reports.

Any medical images contain embedded text, which might identify patient's data, and separating the two independently is often an uneasy task. One approach is although manual, where human intervention allows to filter text data and refill the image, however that exposes the data to the human in action itself. Rather an automated computation is a better solution and alternative which not only has the potential to anonymously complete the task but also process it at greater speed and efficiency.

Although, the challenge lies in ensuring both the quality of the medical image and minimizing the leakage of any sensitive data embedded. However, text may come in mainly two forms:

- printed: follows a rigid vectorized and mathematically shape of each letter, which allows a more sophisticated character recognition and extraction. And in fact, existing solutions are already above satisfactory performance.
- handwritten: it is unique for each individual and does not have a specific pattern, therefore. Often, letters or pair of letters are written in different shapes and curvatures which implies a larger complexity to recognize them. Existing solutions are present, but although still require to discover more potential in greater effectiveness and accuracies.

In this paper, we discuss a project of a planned model pipeline that allows extract both types of texts and replace the text with painted image that resembles as closely as possible to the original image.

MODEL ARCHITECTURE

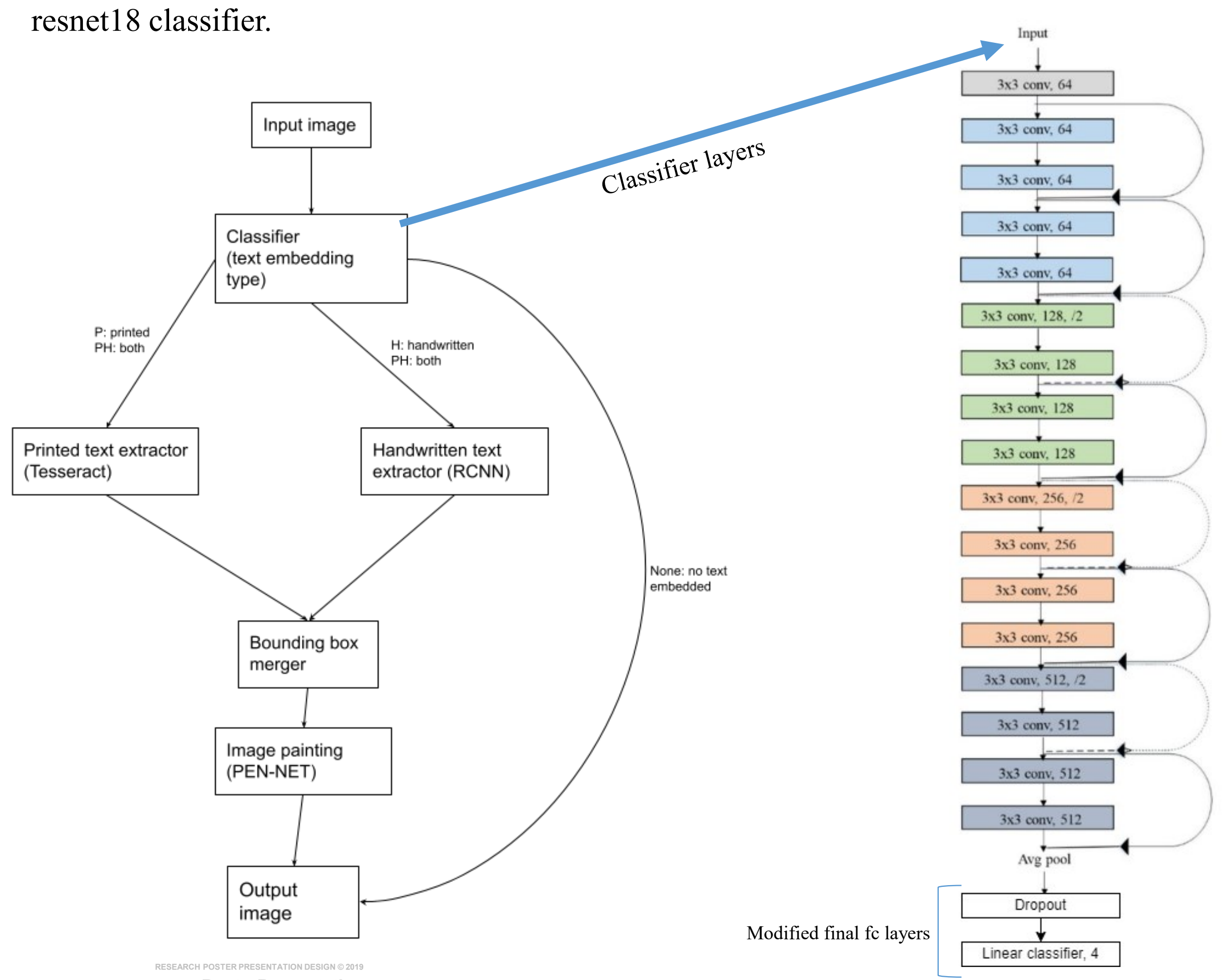
First, the classifier will determine which type of text the image contains. Then, if any text embeddings are present, based on that the image will be processed by printed text extractor and/or by the handwritten text extractor. All the extractors are expected to return the text found and as well the bounding boxes.

Finally, the bounding boxes of the image are replaced and painted by the final layer, which applies image painting. The re-painted image should resemble as close as possible to the actual original image without any text embeddings.

JUSTIFICATION

The classification model is designed with **resnet18**, with the final layer replaced to 4 output classifier (None, P, H, PH). The reason for choosing such a model is since that in order to allow proper feature extraction and then classification of the image, the model needs to be enough deep and complex, but at the same time efficient enough to be effective in practical use.

Model's architecture pipeline. On the right side a detailed structure of the modified resnet18 classifier.



DATASET

First, a publicly available dataset of COVID ct-scans was extracted, which contains original images without any text embedded inside it. Then, for each of these original images, text has been embedded insider them, both printed and handwritten. The embeddings were done by merging text image(s) inside the original one. The background of text images was treated transparently to allow a good level of embedding. If the background of text image was opaque in the embedded result, then the model would falsely behave too accurately as it is easy to find such regions.

For generating printed text images, random sentences strings were converted to images of different sizes in a simple fashion. Whereas handwritten printed text images were pulled from existing IAMS dataset, as unique handwriting styles are simply too many. Thus, in simple words the custom dataset was generated as follows:

- Build printed text dataset
- Build handwritten text dataset
- For each medical image, generate multiple ones with various text embeddings
  - None: no embeddings
  - P: only printed
  - H: only handwritten
  - PH: both

The size of the custom dataset is **approximately 4500 images**, involving all of 4 types of text embeddings (making it diverse) and the source was from SarsCOV2-CTSCAN-dataset from Kaggle, as previously mentioned.

PROCESSING

Each image from the custom dataset was preprocessed to a rescale of fixed resolution (500 x 500), followed by augmentation. On a basis of probability, color jitter and grayscale transformation were applied.

LIMITATIONS

The custom dataset does not contain text embeddings with rotations or other transformation such as shear or curvatures. This is a limitation as the trained model pipeline, if successful, would not be able to extract text accurately from images with such complex embeddings. However, this project's focus is to generate a more viable approach by beginning with a limited complexity level. The **more complex** the dataset becomes the **more difficult** it might be to first assess the model.

TRAINING

PyTorch is the framework used under Google Colab with GPU runtime enabled.

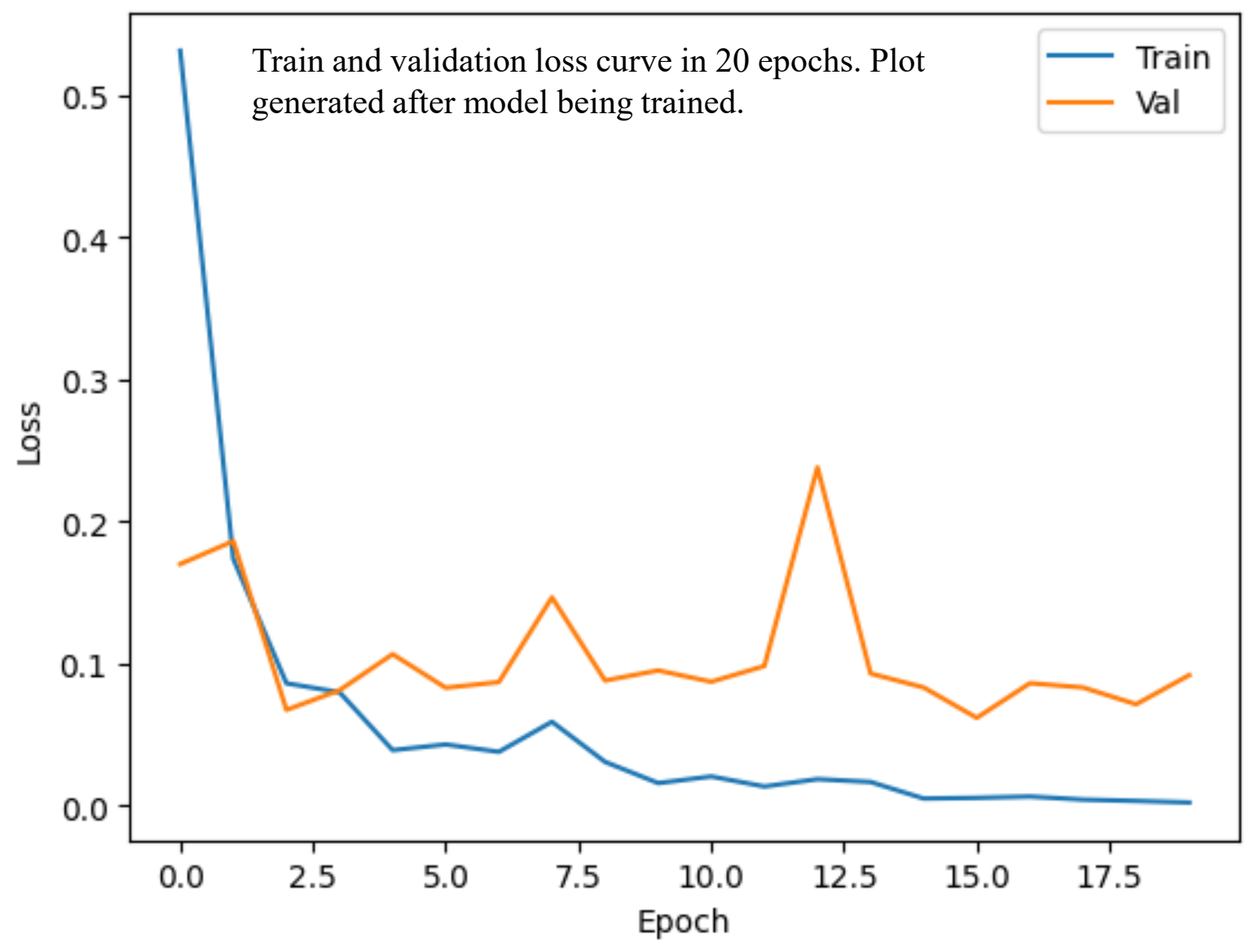
The model used in resnet18, as pretrained, with the final layer modified to 4 output classifier. Initially, transfer learning was applied by only allowing to optimize and change weight of the final layer. However, the accuracies were not sufficient and around 60%, implying the model was too underfitting.

The initial batch size was also low which caused skewness in accuracies between epochs. Thus, the first improvement was increasing the batch size to 32, which allowed the accuracy between epochs to be more stabler and instead of sudden jumps it was continuously improving and increasing.

However, as mentioned the model was still underfitting and thus, all layers of resnet18 were unfrozen and thus allowing to optimize more weights. To **prevent overfitting** in the model, dropout layer was prepended before the classifier, so that the model generalizes well enough.

The optimizer used was SGD with a learning rate of 0.01 and momentum of 0.90. Cross entropy was used to measure the loss function. The results of the classifier with this improvement indeed allowed to achieve accuracies up to 98% after 20 epochs.

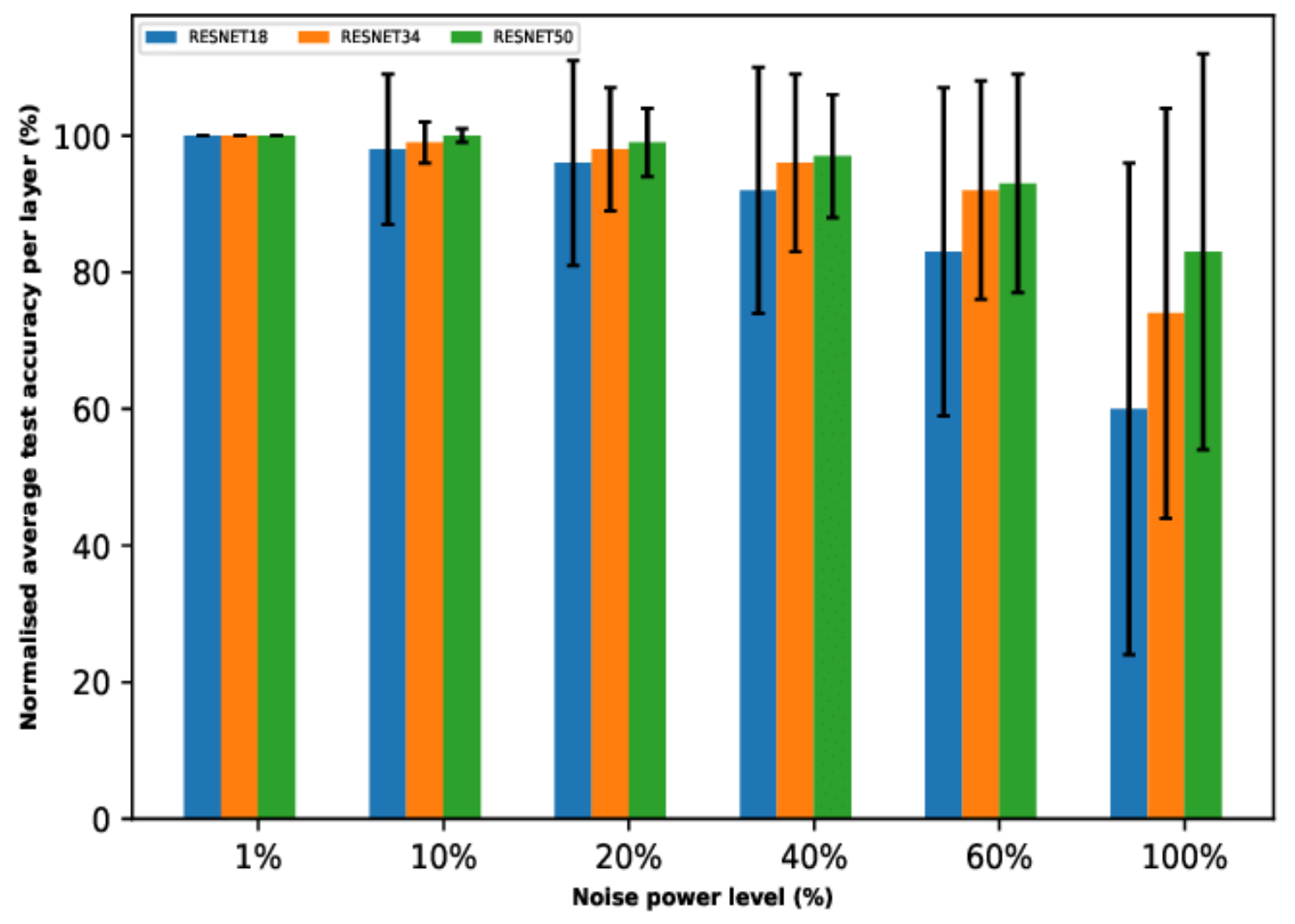
Moreover, it has been noticed that increasing number of epochs beyond 20 (such as 50) did not improve the accuracy and rather the validation accuracy started to drop due to possible increased overfitting.


Train and validation loss curve in 20 epochs. Plot generated after model being trained.

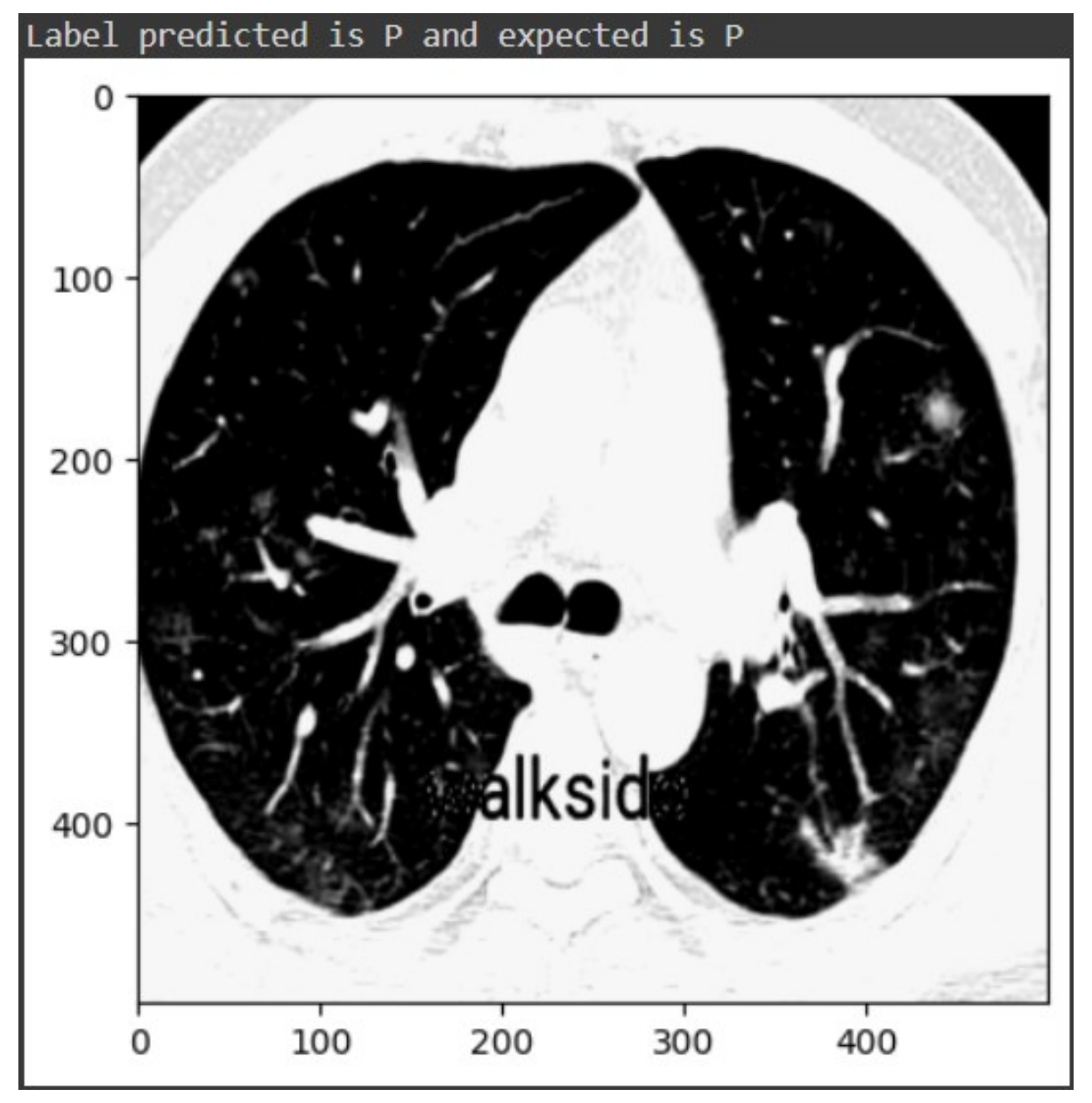
EVALUATION

The accuracy was the evaluation metric used to assess model's performance. Moreover, at the final stage a test dataset was also tested to ensure accuracy of model is consistent with both validation and test dataset which at 98%. In fact, the results of model stay.

Deeper models such as resnet50 were also tested however, the overall accuracy did not exceed 98% as quality of dataset had less noise. The benchmark on the right shows a quantitative comparison of the different resnet models that can be used.



A qualitative example can also be seen below of seeing the result of model of sample input image from the test dataset.


Label predicted is P and expected is P

The sample input image contains only printed text embeddings.
And therefore, P is the label expected, which is exactly the as the predicted one.

MODEL LIMITATIONS AND FUTURE WORKS

Regarding the classifier again, the dataset with text embedding may possibly cause model to sometimes predict wrong value. The main reason is when very little characters are present in the embedded image, which is hard for the model and human itself to trace. For example, a comma, punctuation signs may cause sometimes model to predict as no text embeddings are present.

However, one possible improvement to allow model to achieve better accuracies is to remove from the dataset such image embeddings with minimal text or characters.

The first checkpoint has been reached in the model pipeline and future works involve adding handwritten and printed model, followed by image painting. As previously mentioned, the classifier and all model stages assume text embedded is rectangular and in a bounding box, without any other possible shapes such as circles, curves etc.. The main priority is to create and assess a mode for simpler subset of dataset and then over time in the future newer features can be added to allow model to support a more complex set of dataset.

DISCUSSION

RESULTS

Comparing to existing work of a classifier [Garlapati and Chalamala 2017], [Zagoris et al. 2014], the accuracy indeed matches around 97-98% (using the IAM dataset). This is almost as close with a marginal difference of the results and within a few epochs and short amount of time.

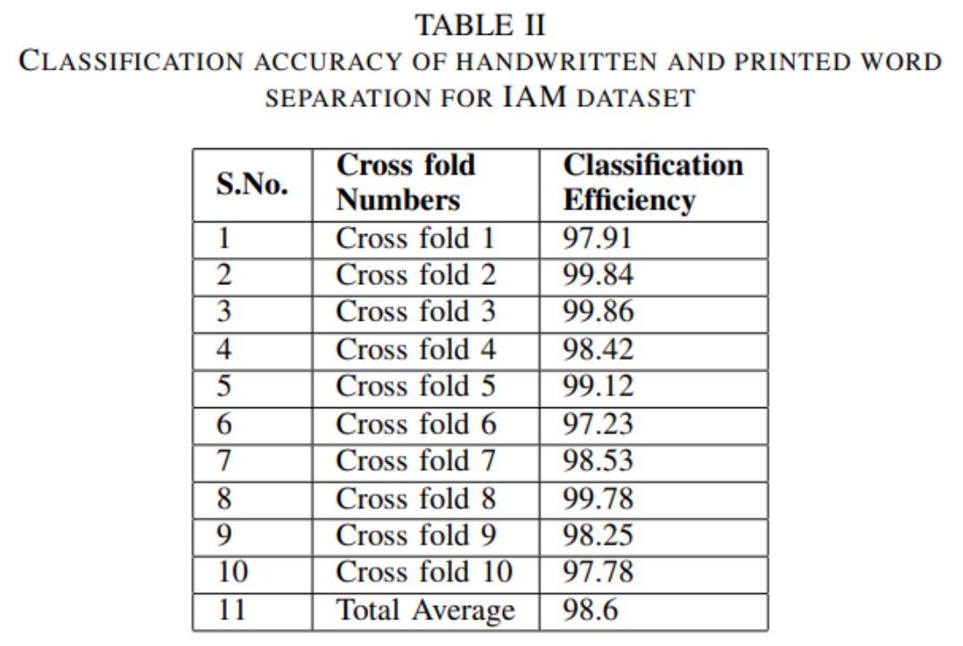
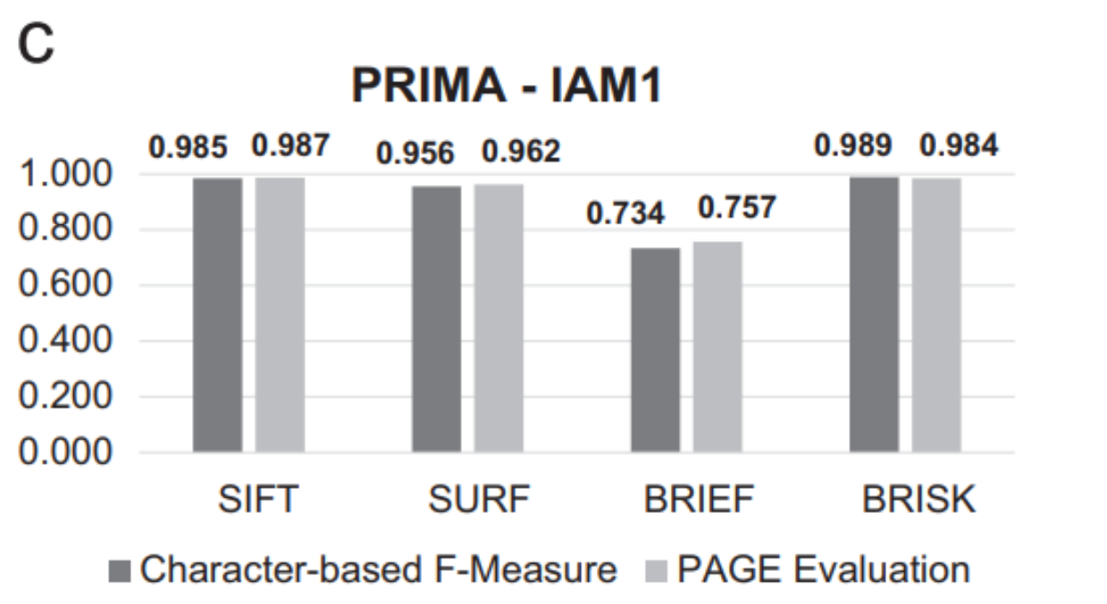


TABLE II
CLASSIFICATION ACCURACY OF HANDWRITTEN AND PRINTED WORD SEPARATION FOR IAM DATASET

S.No.	Cross fold Numbers	Classification Efficiency
1	Cross fold 1	97.91
2	Cross fold 2	99.84
3	Cross fold 3	99.86
4	Cross fold 4	98.42
5	Cross fold 5	99.12
6	Cross fold 6	97.23
7	Cross fold 7	98.53
8	Cross fold 8	99.78
9	Cross fold 9	98.25
10	Cross fold 10	97.78
11	Total Average	98.6

Classifier results of Garlapati and Chalamala 2017



Classifier results of Zagoris et al. 2014

The implications of this high accuracy rate in the context of computer vision are:

- **Increased Efficiency**: The high accuracy rate of the classifier means that you can save time and resources that would have been required for manual sorting of printed and handwritten images. This can be especially valuable in medical contexts where time is of the essence.
- **Improved Data Quality**: By accurately identifying images with printed or handwritten text, the classifier can help ensure that subsequent analyses are based on high-quality data. This can lead to more accurate and reliable results.
- **Potential for Automation**: The high accuracy rate of the classifier also suggests that there may be potential for further automation of the OCR pipeline. For example, if the pipeline is able to accurately classify images as printed or handwritten, it may be possible to automatically select the appropriate OCR algorithm for each image.
- **Future Development**: If the pipeline's high accuracy rate is successful then it may also open up opportunities for further development and improvement. For example, if the pipeline is consistently able to accurately classify images as printed or handwritten, it may be possible to train the classifier to recognize other characteristics of the text (such as language, font, or size). These are exactly the successive steps in building a better pipeline on top of the current one.

FURTHER RESEARCH OR APPLICATIONS

Each user's handwriting style is different and thus different patterns exist. Further research in **determining different types of handwritten text** can allow to build a more complex but effective model pipeline for handwritten text extraction. For instance, once handwritten text is classified, another classifier can be used to determine the specific type of handwritten text so that it text is extracted by a specific model.

One concern, might be the **conflicting cases** where different models extract to **different** text on **same/overlapping** bounding box regions. This indeed seems to be a major limitation and potential issue for the proposed model pipeline, as the bounding box merger cannot distinguish between which of the 2 text is correct. One proposed solution is to indeed re-use the classifier within the image delimited by the bounding box to determine the most likely correct text extraction. Therefore, researching into resolving possible conflicts is another crucial part to make the model pipeline more reliable.

REFERENCES

Bala Mallikarjunarao Garlapati and Srinivasa Rao Chalamala. 2017. A System for
– Handwritten and Printed Text Classification. In 2017 UKSim-AMSS 19th International
– Conference on Computer Modelling & Simulation (UKSim). 50–54. https://doi.org/10.
– 1109/UKSim.2017.37

Konstantinos Zagoris, Ioannis Pratikakis, Apostolos Antonacopoulos, Basilis Gatos, and
– Nikos Papamarkos. 2014. Distinction between handwritten and machine-printed
– text based on the bag of visual words model. Pattern Recognition 47 (03 2014),
– 1051–1062. https://doi.org/10.1016/j.patcog.2013.09.005