Optical Character Recognition for Textual Information Extraction from images of Medical Records and Health Consultation

Yanzhi Wang 1001827416

Mohit Kulkarni 1002031021

Manish Karki 1001852229

Abul Hasan Sayeed 1001911501

Abstract:

Since the start of the pandemic, many doctors have switched to working from home and working online. This can cause problems in collaboration with the patients as many of them are switching to working online for the first time. Therefore, creating the medical report system is gaining tremendous attention in the computer science community and healthcare industry in recent years. We would like to develop an optical character recognition system that can help patients turn printed medical records into digital records quickly and efficiently. The system is able to extract textual information from an image of medical records and submit the info to the hospital system. Specifically, it allows patients to share accurate information with the doctor. On the other hand, it also allows the doctor to remotely access patients’ medical records electronically. So, the doctor can provide helpful advice to patients. In addition to that, we suggest some health advice for patient

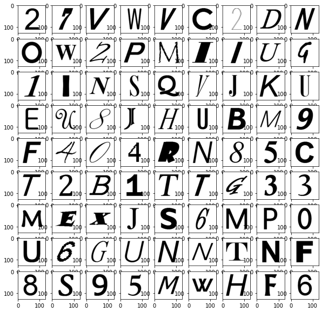
Introduction:

Optical character recognition is the process of turning images of typed, handwritten, or printed text into the machine-encoded text from a document that has been scanned or photographed. It has been used mainly in banking, logistics, and tourism. Now with this project, we are using it in the health sector too. Convolutional Neural Network (CNN), which is a popular deep neural network architecture is used for this project. The LeNet-5 CNN architecture is used for this project. LeNet-5 is capable of learning the important 2D features present in the images and classifying them, the classification is performed by using a soft-max layer. The ability to expedite the data entry process is one of its most immediately noticeable advantages. OCR can assist and shorten the time needed to enter data into electronic health records by automating the data entry process. The ability to increase data entry accuracy is another advantage of OCR. it will help in keeping a proper medical history of the patient. In this project we are taking a new step in OCR health care, We are trying to suggest some remedies by looking at the report of the patient. As of now, when the patient gets the lab report from the hospital they have to wait and get to see the doctor, to know what their condition is as of right now. But as we know due to the pandemic and other obstacles meeting a doctor as soon as you want is not an option. Hence, our project works on this problem and we are able to give the patient a basic overview of their health condition and some suggestions according to their lab report. For this, we recognize the details in the report using the LeNet-5 model, then in the next step, we analyze that data to suggest the patient remedies and inform about their current Health condition. Hence we can say CNN is one of the major inventions of the 20th century, it has limitless uses and it can be very much helpful in the health sector and can save the lives of millions of people.

1. Dataset Description:

We used Chars74K dataset for our training which contains 74k images. It contains a total of 64 classes of English Alphabets and numbers (0-9, A-Z, a-z), it has 7705 characters obtained from natural images, 3410 hand-drawn characters using a tablet PC, 62992 synthesized characters from computer fonts but for simplicity and minimize the error we just used (0-9, A-Z). There was some limitation of the dataset but we did some image preprocessing to alleviate those problems. Some of the image preprocessing that we did were transformation, segmentation, and enhancement enhanced our result. With this, we are able to produce excellent efficiency in our LeNet 5 architecture.

Fig 1: Dataset



1. Project Description

2.1 Description:-

This project will be implementing a model and making desktop software using python and using CNN and TENSFLOW flow libraries. We tested different CNN types and pick the best suited for our project. We experimented with CNN architecture like AlexNet, Lenet-5, RCNN, TableNet and compared their accuracy and training time. From this, we found that Lenet was the best option for us so we implemented it.

After recognizing the individual letter we make the words and give some suggestions to the user with the help of our prediction code. After completing these tasks we will be scanning the medical reports of the patients. Accordingly will be sending the data of the patient which is scanned by the software and give details about the diet and the medications to follow. The system is able to extract textual information from an image of medical records and submit the info to the hospital system. Specifically, it allows patients to share accurate information with the doctor. On the other hand, it also allows the model to help and assist the patient accordingly.

 Fig 2: WorkFlow

Text

Description automatically generated

2.2 Main References for our Project:-

Following are the three papers that we Reference:

1. Xue, Wenyuan & Li, Qingyong & Xue, Qiyuan. (2019). Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2961964.

<https://ieeexplore.ieee.org/document/8941040>

Paper Outline: This paper presents a deep-learning-based approach for textual information extraction from images of medical laboratory reports, which may help physicians solve the data-sharing problem. The approach consists of two modules: text detection and

recognition. In text detection, a patch-based training strategy is applied

1. Dafe, S.G., & Chavhan, S. (2018). Optical Character Recognition Using Image Processing.

<https://www.irjet.net/archives/V5/i3/IRJET-V5I3218.pdf>

Paper Outline: This paper describes the OCR. It's working principles, steps in it. How we should approach an OCR-related problem and different Application

III. Zacharias, Ebin & Teuchler, Martin & Bernier, Bénédicte. (2020). Image Processing Based Scene-Text Detection and Recognition with Tesseract.

<https://www.researchgate.net/publication/340776354_Image_Processing_Based_Scene-Text_Detection_and_Recognition_with_Tesseract>

Paper Outline: This study focuses on identifying and detecting words in real-world photos. The targeted challenge is substantially more difficult than interpreting the text in scanned documents. The availability of images under constraints in the use case under discussion makes it possible to detect the text area in natural scenes.

2.3 Differences in Approach / Method

For this project, we have referred to three reference papers, from all of the papers we got to know various knowledge about the algorithms libraries and many terminologies which helped us complete the project we made. References:

[Paper 1: Text Detection and Recognition for Images of Medical Laboratory Reports With a Deep Learning Approach.](https://ieeexplore.ieee.org/document/8941040)

Summary:

In this paper in order to overcome the data-sharing issue, this work provides a deep learning-based method for textual content extraction  from images of medical laboratory reports. Text detection and recognition are the two modules that make up the technique.

Enhancement:

In this, they have used a patch-based strategy for the detection of high-resolution images. A concentration approach is used in the CNN layers so that it will help text detection and recognition from images of medical laboratory reports. Enhancement: So from this paper we got to know about the CNN algorithm, How the CNN algorithm works as we can see in the above image. This paper uses the R-CNN algorithm for the image scanning part as it takes the Input image then is cropped into layers then goes to the text detector then the detection works and finally we get an output image. The enhancement in this we just got to know the CNN algorithm with this we found that Lenet-5 is the old algorithm which uses the grayscale method and helps us get accurate results as it goes from 7 layers of CNN where the image is shrunk to a smaller version and it gives the accuracy in the grayscale method.

[Reference Paper 2: Image Processing Based Scene-Text Detection and Recognition with Tesseract.](https://www.researchgate.net/publication/340776354_Image_Processing_Based_Scene-Text_Detection_and_Recognition_with_Tesseract)

Summary:

This project focuses on identifying and detecting words in real-world photos. The targeted challenge is substantially more difficult than interpreting the text in scanned documents. The availability of photographs under limits in the use case under discussion makes it possible to detect the text area in natural scenarios. The model is restricted to only a few application situations while having modest computational power needs. In this document, a crucial false positive case scenario that occurred during testing is covered, along with a detailed explanation of the solution. More than 80% of the characters were correctly identified by the project.

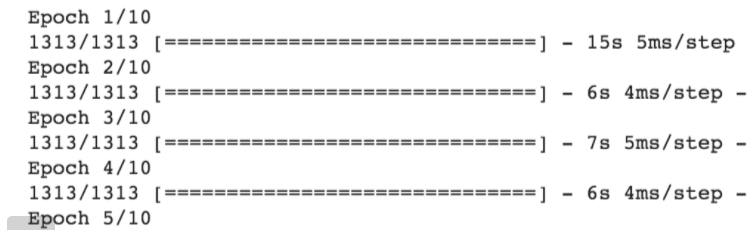
Enhancement:

This project uses Tesseract which is an optical character recognition engine with open-source code. The text lines are tested by Tesseract to see if they have a fixed pitch. When Tesseract encounters fixed pitch text, it uses the pitch to slice the words into characters and turns off the chopper and associator for these words during the word recognition stage. Tesseract, however, has significant limitations. The OCR is not as precise as some of the commercial options we have at our disposal. Doesn't work well with pictures that have artifacts, such as partial occlusion, warped perspective, and complex backgrounds. It is unable to discern handwriting.

To overcome these we used Lenet-5 because it can make good use of the structural information of images, it has low training time and accuracy is pretty high.

2.4 Differences in Accuracy/Performance

We compared the accuracy of different Architectures along with changing different hyperparameters. We changed epoch no, and batch size and calculated the Performance of different architectures.

The following Figures demonstrate the time required to run each epoch.

Chart, line chart

Description automatically generatedText

Description automatically generatedChart, line chart

Description automatically generatedFig 3: Lenet-5 Epoch Run Time

Fig 4: RCNN Epoch Run Time

From the above figure, we can see that Each epoch takes much more running time in RCNN than on Lenet-5. The time difference between each time taken per step was huge but there was no significant change in the accuracy. This was the main reason that we used Lenet instead of RCNN. As the result was the same will comparing Lenet with RCNN. The accuracy change was not that different but the running time was significantly different.

3 Analysis

In the first phase of the project, we were going to implement the project by creating a scanner software using tesseract and TensorFlow libraries so we started researching and getting reference papers that can help us understand the main concepts and terminologies of the TensorFlow libraries and tesseract packages. Then we came across a paper with Image Processing Based Scene-Text Detection and Recognition with Tesseract, where we got to know about the tesseract and how it works. The overflow of how the images go from preprocessing to the text area detection and by image cropping and brightness check and gamma reaction helps to make the text look brighter so that the text is extracted properly. In this paper, they used Tesseract V5. So we started implementing the scanner application in tesseract and modifying it with more gradient scale methods so that the accuracy of the text detection is perfect. So in the paper, we came to know that the accuracy of the text detection is overall 83 percentage and the overall character accuracy is 83.7 %. We wanted more accuracy as we have to determine the patient's illness and suggest to them the meal plan according to their symptoms. So, we moved on to the first presentation where Professor suggested we look out for Conventional Neural Network algorithms. After searching some algorithms, we found out about Alex Net Conventional Neural Network. As we started Implementing the project in Alex Net, we came to know that computational power and time consumption are more in Alex Net. Then we came across a similar paper to our project Text Detection and Recognition for Images of Medical Laboratory Reports with a Deep Learning Approach Where they used the RCNN algorithm which is the same as Alex net with High computational power and time consumption. Then we started implementing the project in Lenet5 and we got the same accuracy as the Alex Net and RCNN but in less time. As you can see in the figure for AlexNet time consumption and computational power.

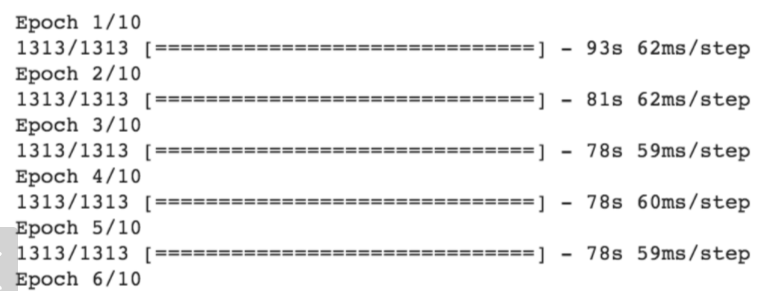


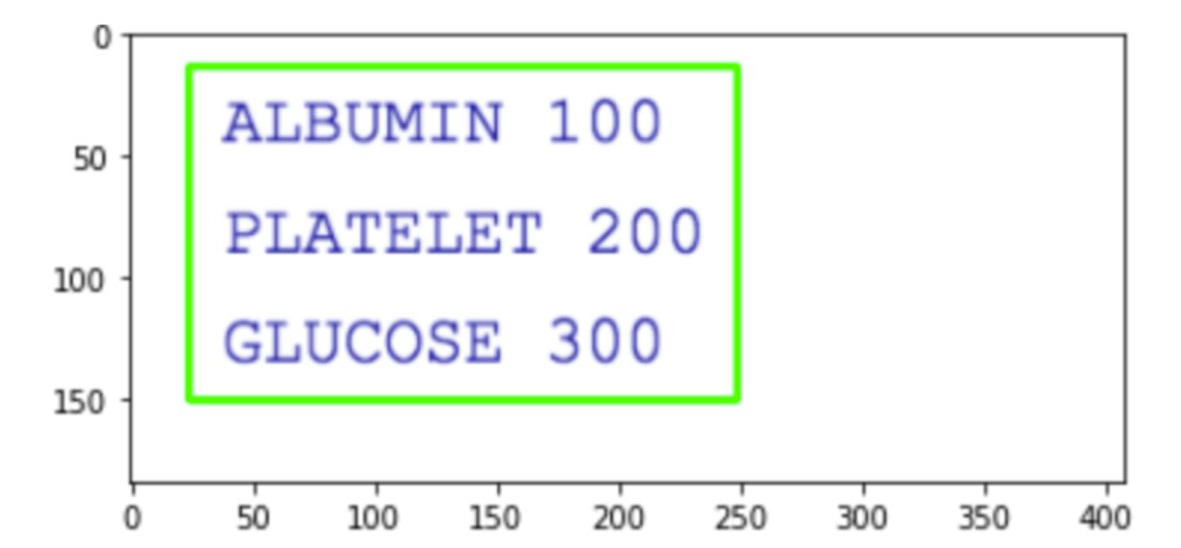
Fig 6: Alex net 10 epochs iterations for the Dataset

First creating the Lenet5 CNN model by implementing the algorithm and recognizing the words in every line. We got accuracy with compare to our dataset which we used for training the model so that we can get the accuracy of the model we created. So we were able to detect the text from the datasets but there was a problem we were facing. There is a large space between each line and we cannot change the word size and style. The algorithm is that we recognize

each line in the image. After that, we recognize the word in each line. After that, we recognize the character in each word. Therefore, we have to set up the threshold. If characters are within the threshold distance, they are grouped together. Otherwise, the characters are not in the same group. In this way, we can put the nearby character in the bounding box in order to generate the word. Therefore, we have to ensure that there is a large space on the left side, right side, below, and above. We cannot put the word in the table because it is going to recognize the table object as a character. So after solving the problems by resizing the images and cutting out pixels we were able to decrease the spaces between the characters.

Table

Description automatically generated

Fig 7: Limitations and Problems faced during the implementation

As we were able to get the characters from the dataset we started implementing them on the dummy medical reports and we were getting accurate results. For the prediction part, we first use the BMI calculation function with the help of the height and weight of the person. then it checks if the BMI is less than 18.5 then it tags you in the underweight section and what all diet you should follow. Likewise, it states for the BMI is normal from the 18.5 to 25 range. After that, it states for the overweight which ranges from 25 - 30. And it categorizes the Obese for ranges greater than 30 BMI. Then we focused on the RBC and WBC counts and measure them according to age and state whether your hemoglobin is normal which ranges from 11.5 to 15.5 in your body, is it low which ranges less than 11.5 and you have to eat fruits vegetables, etc. At the last Is the Hemoglobin high if it exceeds the range? Then It has a function of RBC with normal ranging from 4.0 to 5.5, low ranging from less than 4.0, and high RBC ranging from greater than 5.5. Then comes the WBC count with WBC normal between 5000 - 10000, low ranging from less than 5000, and high ranging from greater than 10000. At the last, we made some more Enhancements by calculating the Hemoglobin, WBC, and RBC as Per the Gender

So Male gender range for Hemoglobin is 13.5-17.5 for normal hemoglobin, less than 13.5 for low hemoglobin, and

greater than 17.5 for higher hemoglobin following every diet for what to take and what not to take. Same for the RBC ranging from 4.7-6.1 for normal, less than 4.7 for lower, and higher ranging from greater than 6.1. Now comes the WBC count with normal having 5000 - 10000, lower ranging from less than 5000, and higher ranging for greater than 10000 After each calculation we have given a specific diet to follow for the categories. Likewise, we have implemented it for the Female Gender.

**3.1 What did I do well?**

* We had high accuracy while both training and testing of 99.3%
* We compared 3 different CNN architecture
* Compared to the reference papers we were successful in implementing Lenet5 architecture instead of using RCNN. As we got results the same as RCNN  each step in the epoch for LeNet-5 is 5 seconds and each epoch for RCNN is 90 seconds. So, it can prove the time it takes is much longer.
* Finding the optimum hyperparameter for my CNN architecture

**3.2 What could I have done better?**

* If we were able to recognize the words inside the table more accurately then its application would be more diverse
* Using another model to predict the Health Issues on the data from the first model would be a better project
* If we were successful to implement the new CNN architecture TableNet that would have made the project better
* We should have used higher configuration desktops or Pc so that we could have gotten accurate results in less time for other architecture.

**3.3 What is left for future work?**

* We can integrate another model to predict health issues from the first model's data
* We can mine the medical report from the internet and train our model for more accuracy
* In future work, we can implement more applications like suggesting to the patients with the particular doctors which are available in their local areas and sending the contact details to them.
* Also, give them directions on google maps where the doctor's clinic or the hospital is there so that the patient can visit and check their symptoms.

4 Conclusion:

We were successful in implementing a CNN architecture for OCR and extracting text from a Medical Report and were able to suggest to the patient according to the information in the report in screen output or output file that can be printed. Below we have a screenshot of the input Medical Report and output that our program provided.

Fig: Output

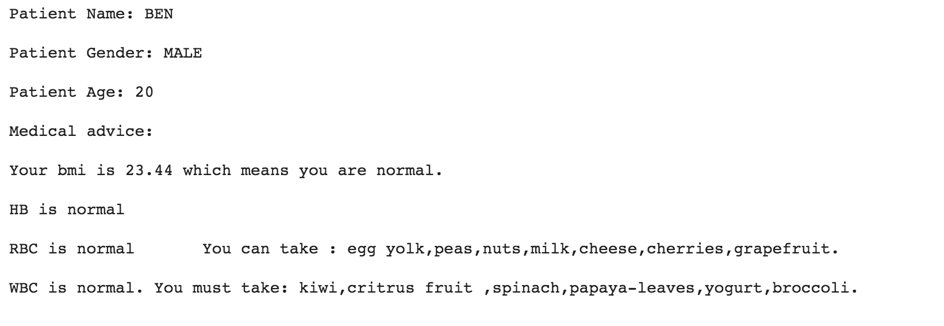


 Fig: Input Lab Report