

Document Text and Template Identification and Data Extraction Using Machine Learning And OCR On Android Environment

Name And ID:

1. Name: Md. Minhazul Islam Rimon  
   ID: 20101078
2. Name: Md. Fuad Islam

ID: 20101060

1. Name: Tasnim Mobarak  
   ID: 20101296
2. Kaushik Roy  
   ID: 20101185
3. Mysha Samiha Priota  
   ID: 20301205

**Supervisor: Dr. Md. Khalilur Rahman**

**Co-supervisor:**

**Date of Submission:**

**Department of Computer Science and Engineering**

**BRAC University**

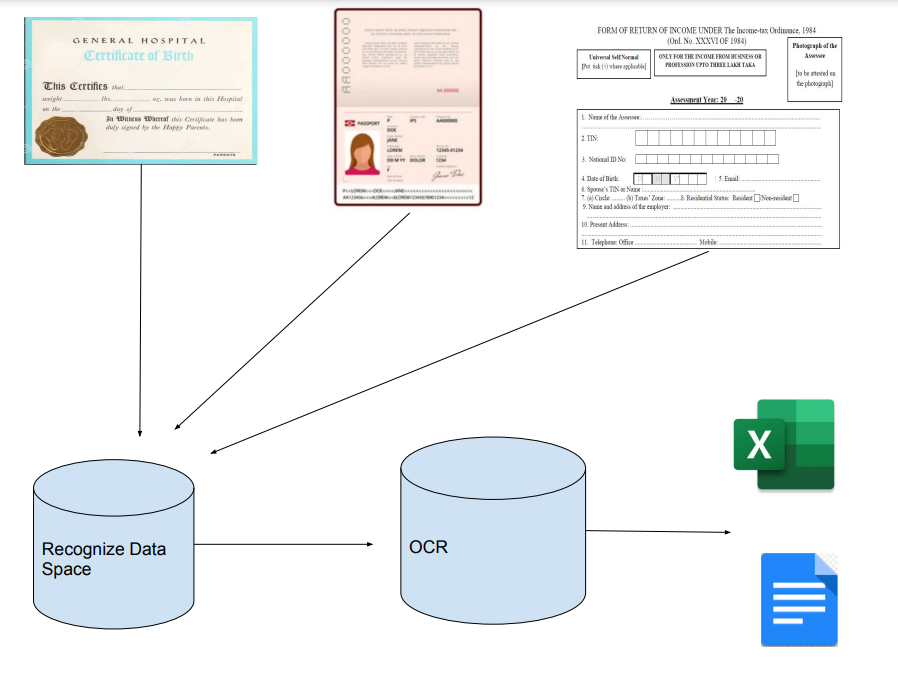
Document Text and Template Identification and Data Extraction Using Machine Learning And OCR On Android Environment

**Abstract**

At present, we are living in a technical world in which digital devices are available to everyone. Be it advanced computers and tablets or in general smartphones, we even surpassed the age restriction and now toddler and pre-school children can somewhat understand smartphones. Even though we have the opportunity to use these devices as a medium to fill official documents and use it to carry virtual identification documents, we are still stuck with obsolete paper based documents. However, when it comes to sending the documents for official reasons, we have to scan and send them. Then comes the most manual part of the process of identifying the data of the documents and putting it in the system. We wish to automate this part of the process for better efficiency and remove the scope of error. As most people have access to a smartphone, we want to apply a machine learning model in an android environment so that it can identify documents and its templates with different sizes with data scattered in unique fashion for each of them as the person is using it and suggest recommendations. After identifying the area of text, we wish to use OCR - Optical Character Recognition to identify the text and upload it to a suitable location. This way we can save time by shortening three stages of processing, data uploading and verifying into one step of taking a picture of the document.

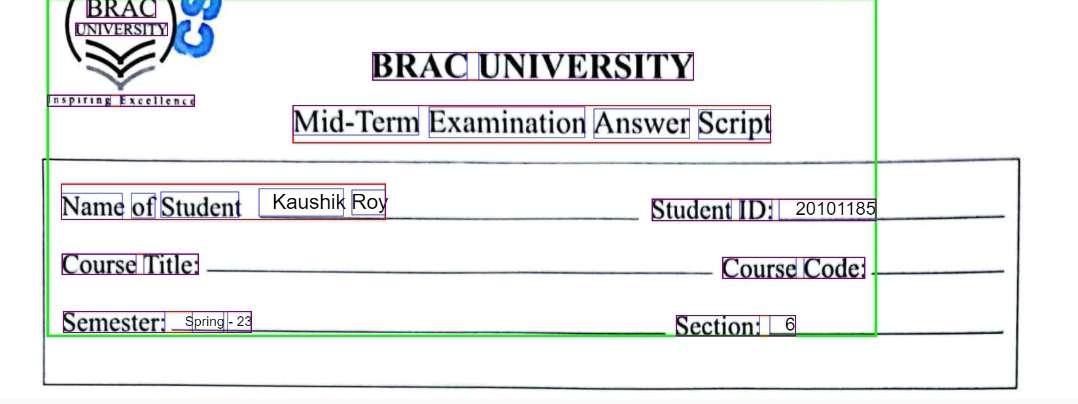
1. **Introduction**

In our daily life, we go through and process an impressive amount of documents. Documents like birth certificates, income tax, different types of government forms, passports, exam papers etc. Almost in all cases, we are required to extract data from these documents and copy them to a different virtual storage. These processes are tedious and time consuming. And in more than enough cases, there are bound to be some mistakes made which makes them even more obscure. That's why we need a system that will automate these processes and make these tasks more efficient.



**Figure 1:** *General Process of the Application.*

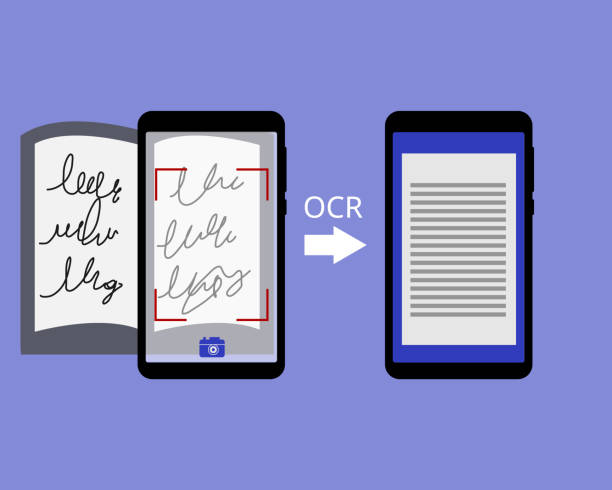
The first obstacle that we need to overcome is identifying the document. Each type of document has their own characteristics and there are specific places from where we need to extract our data. We are approaching this problem with a two step process. Before extracting data from any type of document, we will ask the user to make a custom template where they will take a test picture and highlight the pieces from where data needs to be extracted. Behind these processes, we will have a Machine Learning model which will analyze the user behavior and learn the pattern of places where important data are stored. So, when the next time the user wants to create a template again, this model will give the user recommendation based on the previous cases. This ML process of test case and test data will occur simultaneously with the user using the system and it will be done offline. By being offline, we can ensure that a large number of users who do not have immediate access to the internet can use the service whenever necessary.





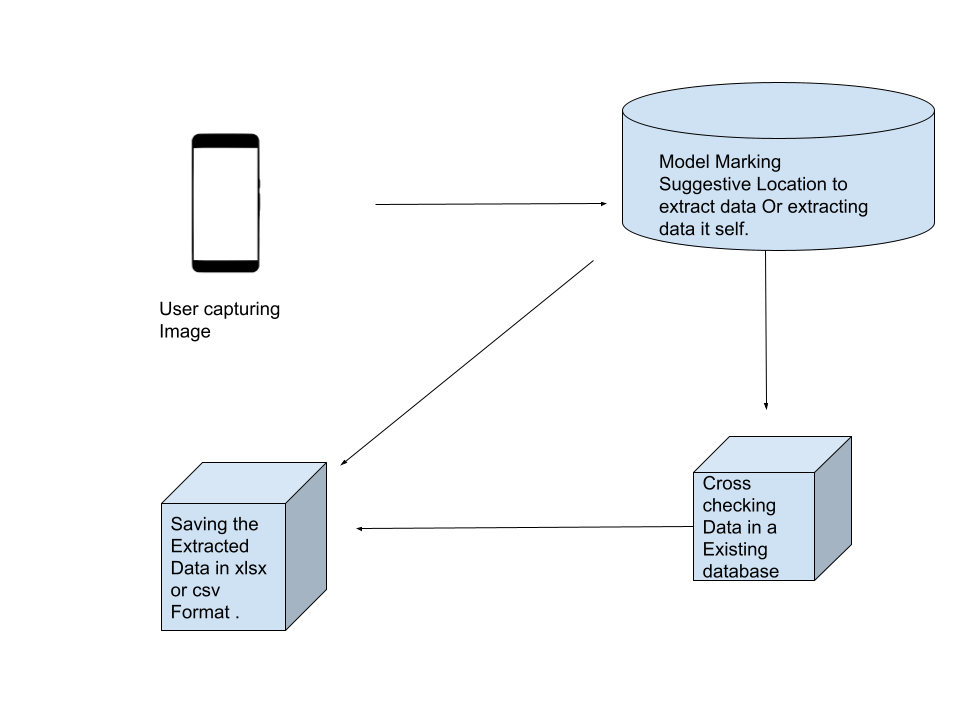
**Figure 2:** *Bounding-box on around the texts.*

For the data extraction part we will use OCR. OCR is currently one of the most researched topics as even if there are quite a few models on the market. None of them are able to achieve 100% accuracy. There are Google's cloud vision API, IBM datacap, Adobe readers etc. These software scans the written or printed paper characters, processes them and converts them to machine readable text. The available models that we discussed are available only on computer devices or doesnt work offline. For the convenience of the users and lift the restriction, we want to preprocess them locally in the android devices. For this we need to scan the documents first, then preprocess it to fit for extracting, then extract and classify it into machine readable texts. Then we show the output as classified text of the document.



**Figure 3:** *OCR ( Optical Character Recognition )*

Finally uploading the extracted data into a suitable file. For this simple step we can add an algorithm to upload data to a local file or user connected file on google drive for user convenience. In addition to these, there are times when we have the option to check the validity of the data from an already existing database. For example: correct fathers name from old passport or birth certificate, correct tax id from old tax reports etc. In these cases, we can add a dynamic program which will crosscheck the extracted data from the existing database. When partial encounters of discrepancies in unique keys, it can alert users for checkup.

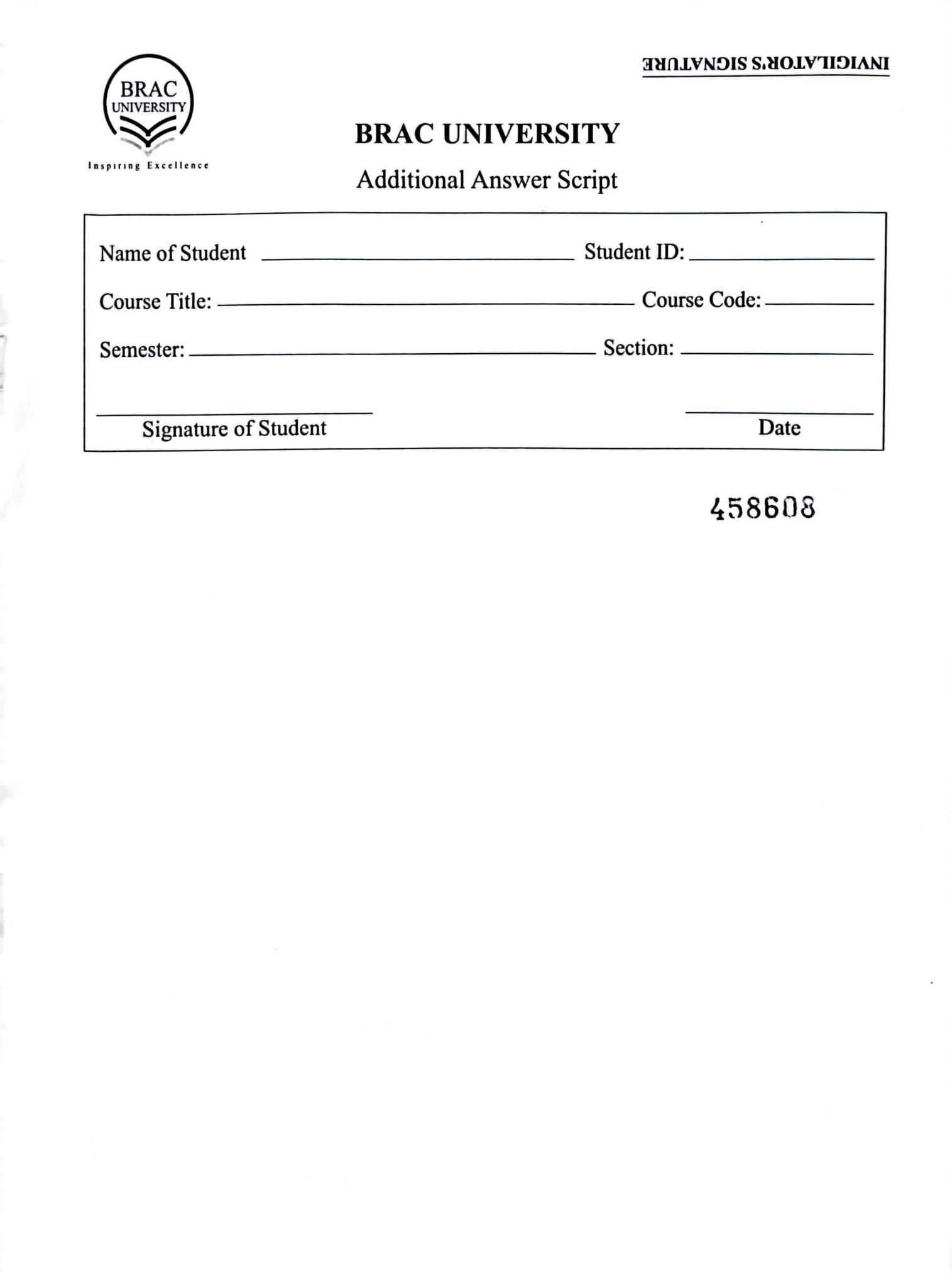
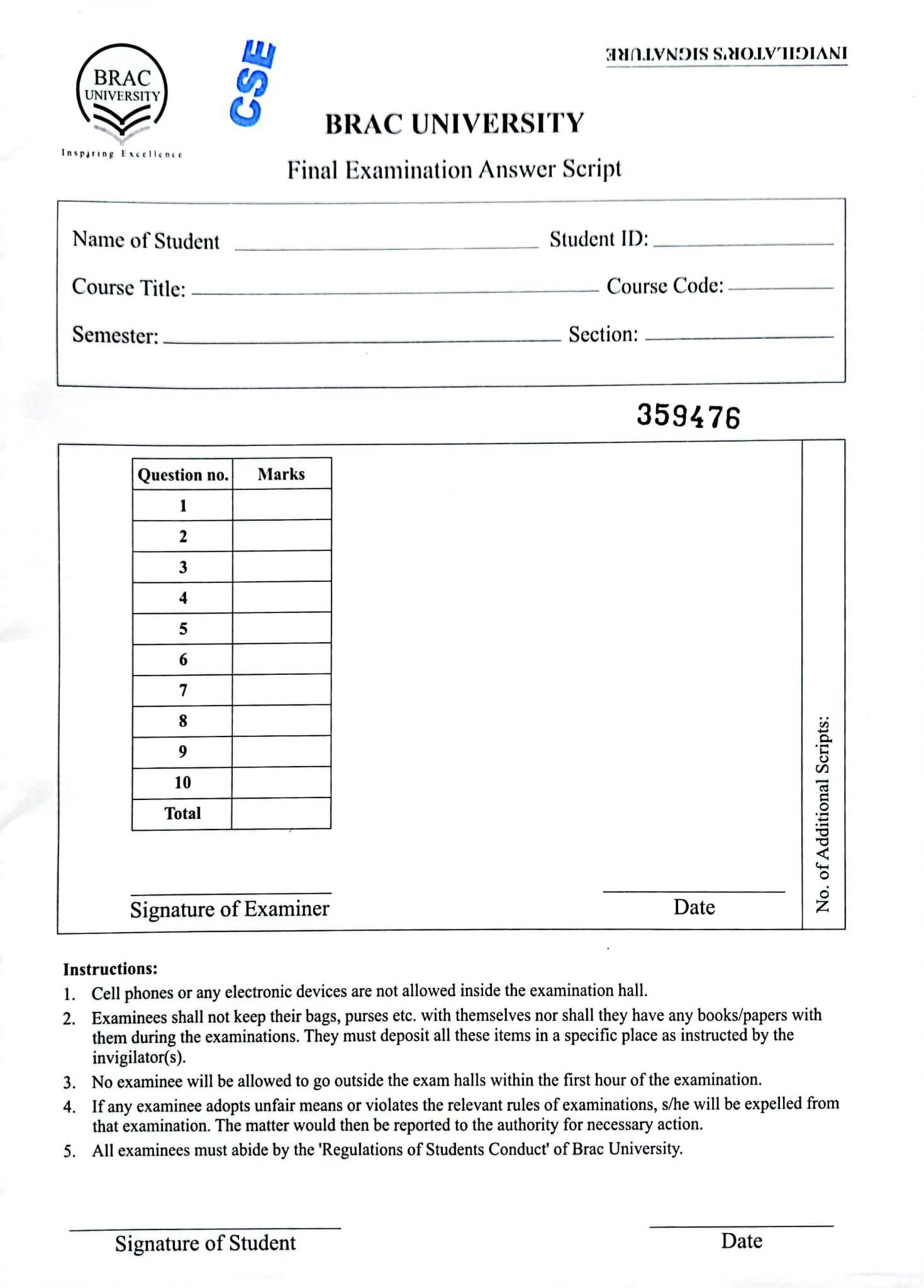
****

**Figure 4:** *Steps of working process of the Application.*

This three phase model aims to increase the efficiency of extracting data and improve the accuracy of existing OCR models. In addition to this, this will give users an easy to use and access environment of the day to day task. These report aims to discuss about following topics:

1. A detailed description of the current situation of the OCR models
2. Advantages and disadvantages of shifting to mobile device for these task instead of specific gadgets
3. Restrictions faced by past models and how we aim to overcome them
4. Suggestion of implementing Machine Learning algorithms in any android environment and effects
5. **Research Problem**

Nowadays everything is digitized and any information can be easily accessed if it is in document form. Every day we come across different types of documents which we often take pictures to preserve. Such documents include driving license, passport, national identity card, exam papers etc.The texts in these are written using a wide array of letters, numbers, and symbols. But if we could collect the information from such different pictures and keep it documented in one place, it would be convenient to arrange it as well as it can be found quickly when needed. Our goal is to to detect these distinct information from a picture and input them straight into an excel sheet.



**Figure 5:** *Front Page of Exam Paper which will be used as template.*

In every educational setting, evaluating exam papers is a crucial task. Keeping track of the number the student received during their exam at a later date is likewise a time-consuming chore. The process of recording exam results is getting more and more time-consuming due to the rate at which the enrollment in educational institutions, particularly universities, has increased in recent years. Various types of sections can be found on an exam paper. Some of which include the student's name, certain details about his class, or some contain information about the grades he earned. All of these distinct pieces of data must be included in one document in order to maintain a record of a student's exam results. Our goal is to simplify this task through the findings of our research on the subject. When reviewing exam papers, the most frequent issue teachers encounter is having to spend a lot of time filling out an excel sheet with all of the student information. The final outcome is delayed significantly as a result of the lengthy process of uploading each student's grades into an excel sheet. We intend to automate this manual approach and reduce computational overhead while maintaining excellent performance and accuracy in document detection and data extraction.

Using machine learning and OCR to address the aforementioned issues might be a practical approach. Our goal is to include these methods into user-friendly software in the Android environment. But implementing such an approach involves several challenges. Because not all students have identical handwriting, this activity presents a number of difficulties.The way that each student writes letters and numbers may be different, thus there will be unique writing styles. The system would then need to be trained on various handwriting styles in order to recognize the same letter or number. Our objective is to reliably distinguish letters and numbers from various styles of handwriting.

1. **Research Objective**

This research aims to create an android application which will enable the users to scan and extract data from any kind of document with different kinds of templates ex: passport, certificate, government form etc. It will try to utilize the Android environment to create a ML and OCR model which will process the data accordingly. So the objectives with this model are:

1. Enable the user to extract data from scanned documents and save them to users convenience
2. Give user recommendation to the places from which data can be extracted
3. Upload the extracted data to convenient online or offline database or excel file
4. Crosscheck data from existing database (when available) to ensure data validity and accuracy
5. Enable processes in the local environment to ensure reliability when the internet is not available.

**II. Literature review**

Shubham et al. [1] used CNN (convolutional neural network) model for recognizing handwritten texts. They used the Eminst dataset for both training and testing purposes. In their approach, they first pre-processed the data via normalization, rotating and reversing, input image filtering. With data processed and fit for training, they used the SEQUENTIAL model which consisted of 8 linear stacks of layers. Lastly, they used various optimizers available for keras to train their model. After experimenting with all the models,they found out Adamax, a first order gradient-based optimization method provided the most accurate results.Also they used 2 convolutional layers for maximizing accuracy. With Adamax, the trained model had an accuracy of 87,1 percent. Though other optimizers like Adadelta, Adam, SGD provided close enough accuracy, Adamax reduced the training time significantly. With better data pre-processing methodology the accuracy, efficiency of this model can be improved and for our work using a 2 convolutional layers model with Adamax optimizer provided by keras is a viable option.

Shruti et al. [2] used semantic segmentation and other pre-processing methodology to recognize mixed data (handwritten, printed texts) faster. In their work, they prioritize pre-processing data over improving conventional OCR models. Their approach focused on using any OCR engine readily available and getting optimal results by inputting fit data for training which they would get from their elaborate pre-processing techniques. First the input image will be pre-processed where binarization, noise reduction, slant removal, text alignment issues were performed. The resultant image was then processed using Liner removal, Gray Scale Conversion, Gaussian Blurring, Thresholding, 3 Channel Re-Conversion. Afterward, using U-net image segmentation was performed and on the output using OCR engines digital text was generated from handwritten/printed text. In their image processing module, there was a label isolation model that allowed them to segregate handwritten texts from printed text. This module can be used in our research for using segregated printed text for recognizing and categorizing different labels. Also, their pre-processing approach is also very viable for our approach as we want to recognize texts into different segments based on labels. Their work focused on forms which are an example of mixed data while our work focuses on exam transcripts information page where student information and grading is provided in mixed data as well.

Zhengchao et al.[3] combined CNN and RNN to recognize scene text. CNN was used to extract features from an input image. Different descriptors like VGG16, VGG19, ResNet34, ResNet50 were used for feature extraction. Now extracted data was sequential feature map and CNN has a disadvantage in such cases. This is where RNN which can extract sequential objects of arbitrary lengths comes in. After experimenting with different neural networks they managed to combine advantages of both CNN and RNN and claimed that a deeper CNN with deep descriptor will be more effective in predicting scene texts.

Yi Jiang et al.[4] proposed a text recognition algorithm to solve text segmentation difficulties, dictionary dependence using Attention mechanism and connection time classification.Here, low resolution images are upgraded using ESPCN into high resolution images. Afterward via multi-scale CNN, features are extracted. Later using Attention - CTC said encoded feature sequences are decoded. Traditional RCNN has great success in recognizing text but the training takes a significant amount of time and the calculations are large. Via the algorithm identifying the target is simpler and takes less time and calculation. In their work, they came to the conclusion that training models based on Attention CTC avoids the gradient disappearance problem of RNN. Proposed algorithm here can be advantageous if it can be implemented on android devices as effectively reducing time needed to operate would lessen the burden on the hardware.

Again, Himank et al. [5] proposed faster and efficient pre-processing methodology. Here geometric rectification, pre- processing, image detection, text extractions are key elements that are used for recognizing text data. Findings of their experiments suggest that though their approach reduces the time complexity and simplifies the issue at hand for untrained text fonts the approach seems to be lacking. However, with trained text fonts the accuracy rate of successfully digitizing texts is around 80 percent. We can improve the OCR model to make it adaptable to untrained fonts and use the proposed methodology in our endeavor.

For scene text recognition, Xia et al. (2022) [6] suggested a Transformer-based encoder-decoder structure with a two-stage attention mechanism. A first-stage attention module that combines spatial attention and channel attention captures the overall placement of the text in the image at the encoder, and a second-stage attention module at the decoder precisely determines the position of each letter in the text image.This study suggests that this two-stage based strategy can improve recognition precision and more precisely determine the position of the text. In order to provide more robust features for the encoder, they also developed a multi-branch feature fusion module. This module can merge features from several receptive fields. According to this study, this framework is better for learning than the RNN-based STR approach and can speed up training. The encoder of this framework uses convolutional layers instead of linear ones because the input is an image, but it also adds that the original Transformer encoder's structure is kept.

In their work,Panchal et al. (2022) [7], investigated how Android's image recognition technology can extract text and features from images. They studied acceptable algorithms and their accuracy, which is a vital component of each paper, by using a variety of research papers from other academics. Every research paper was examined in the context of Android technology. Utilizing the earlier recommended methodologies and algorithms, feature and text extraction work has reached 85% to 88% accuracy as of 2012. Since Android has been widely used since day one, using an android application makes this analysis more authentic. The paper concludes by suggesting that even though there are many methods and APIs accessible, there isn't one single, universally applicable solution because it relies on the application's requirements.

Anarghya et al. (2020) [8], worked on stroke detection and Hog transformation method. Their strategy presents a method towards character identification and recognition that combines the advantages of feature extraction methods associated with components distribution. The algorithm focuses on the clarity of text background segmentation. After reviewing some research papers and algorithms, they proposed integration of the Hog transformation and stroke detection method to achieve higher accuracy in localization and recognition in OCR technology. This approach highlights its efficiency in dealing with different text patterns, light and shadow conditions, and linguistic issues. Standard datasets have been demonstrated and improvement on previous OCR technology has played a significant role in providing a solution for real-time scene text recognition. This paper used MODI which can create difficulty in getting contents from books.

Jyothi et al. (2020) [9], used OCR model to detect text and separate them into different graphical portions. Their proposed OCR model can classify different identity documents such as, passport, license into different categories. First of all, their model categorized the photos and derived information from the text extraction module and then the identification details were stored in the database. A new neural network based OCR engine LSTM( Long Short- Term Memory) is used to detect character patterns in this research. The research is mainly focused on the main project's purpose, technologies employed, used databases and the functional procedures. But they couldn’t define a best solution for making the algorithm more efficient.

**III. Implementation**

To implement the model for real world scenarios and to maintain accuracy, the steps will be:

1. Template gather: As there are many different types of templates, in order to generalize the pattern of spaces where data are available. We need to gather different types of document templates such as: Birth Certificate, passport, tax document, financial document, exam papers etc.
2. Text data gather: As we want to process our OCR model locally, we want to initialize it with a bunch of hand written and typed characters in different fonts and shapes.
3. Template pre-processing: Before using it into a Machine learning model, we want to process it to cut out unnecessary areas such as logos, images etc.
4. Template processing for model: Now we want to process these templates to find the places for each unique case where the data are situated. Then the model will suggest the user's finding. Based on user response it will correct itself and apply that decision for future reference.
5. OCR pre-processing: Same as template, we want to only use the characters from our data set, extra spaces around and in between should be cut out to improve the accuracy.
6. OCR processing: With the character dataset, we will process them to identify the pattern for the characters and cross reference the output with the existing database (if available). Then show the output to the user and upload to the database.
7. Testing and Evaluation: In order to measure the performance of the model we will use different real world documents for testing. Effectiveness will be measured with appropriate metrics to ensure user satisfaction in real world scenarios.
8. Deployment: With appropriate model design and evaluation with real world tests, it would approximately take 7 months to make it to a feasible phase. Then we can deploy it for a real world environment.
9. Maintenance: The human handwriting is different from one another and it continues to differ as time progresses. To ensure the performance of the system and handle future gimmicks, a scheduled maintenance will be done to make appropriate tweaks. Additionally as user preference is saved in devices, we can gather them as a qualitative research data set to thoroughly understand user preference of text and data space and come up with appropriate solutions for the future.

**IV. Conclusion**

To conclude, integration of ML and OCR in an android environment has great potential as it directly enhances user experience by removing the human labor and makes it more time efficient. In addition, the cross checking feature will rapidly ensure data validity and speed up the process of error handling even more. Then by enabling the processing to occur in the local environment, people can now store data with ease when visiting rural areas with no reception. Furthermore, the model uses user reference to calibrate itself for better results. So, the model can ensure each user gets their own preference in extracting data.

**Reference**

1. Mor, S. S., Solank, S., Gupta, S., Dhingra, S., Jain, M., & Saxena, R. (2019, February). HANDWRITTEN TEXT RECOGNITION: with Deep Learning and Android. *International Journal of Engineering and Advanced Technology (IJEAT)*, *8*(3S), 7. <https://www.ijeat.org/wp-content/uploads/papers/v8i3S/C11730283S19.pdf>
2. Patil, S., Vijayakumar, V., Mahadevkar, S., Athawade, R., Maheshwari, L., Kumbhare, S., Garg, Y., Dharrao, D., Kamat, P., & Kotecha, K. (2022). Enhancing Optical Character Recognition on Images with Mixed Text Using Semantic Segmentation. *Journal of Sensor and Actuator Networks*, *11*(4), 63. <https://doi.org/10.3390/jsan11040063>
3. Lei, Z., Zhao, S., Song, H., & Shen, J. (2018). Scene text recognition using residual convolutional recurrent neural network. *Journal of Machine Vision and Applications*, *29*(5), 861–871. <https://doi.org/10.1007/s00138-018-0942-y>
4. Jiang, Y., Jiang, Z., He, L., & Chen, S. (2022). Text recognition in natural scenes based on deep learning. *Multimedia Tools and Applications*, *81*(8), 10545–10559. <https://doi.org/10.1007/s11042-022-12024-w>
5. Dave, H. (2020). *OCR Text Detector and Audio Convertor*. <https://www.academia.edu/43229530/OCR_Text_Detector_and_Audio_Convertor>
6. Xia, S., Kou, J., Liu, N., & Yin, T. (2022). Scene text recognition based on two-stage attention and multi-branch feature fusion module. *Applied Intelligence*.

<https://doi.org/10.1007/s10489-022-04241-5>

1. Panchal, B. Y., Chauhan, G., Panchal, S. R., & Chaudhari, U. M. (2022). An investigation on feature and text extraction from images using image recognition in Android. *Materials Today: Proceedings*, *51*, 798–802. <https://doi.org/10.1016/j.matpr.2021.06.237>
2. U, A., S, S. U., & Rodrigues, A. (2020, May). Text Localization and Recognition. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *8*(V May 2020). [www.ijraset.com](http://www.ijraset.com)
3. E, J., Tejaswini, K., Chintalapati, L., & Shafiulla, M. M. (2020, May). Text Extraction from Images Using OCR. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *8*(V). [www.ijraset.com](http://www.ijraset.com)