Handwriting Detection System

About-

In today's date everyone is living in a generation where machines are smart enough to make calculations and somewhere around machines are leveraging most of our things. Be it manufacturing ice creams or creating a document to give a presentation. Almost 90% of the things a general person does in today's date somewhere around involves machines and ai into it. During this era, it's really important for the machines to be made smart enough to read our documents or ancient sculptures so that things humans may not have interpreted correctly or may not be used in the best way can be made into use with the smart machines. We all know most of our ancient documents are still preserved in a book or diary where it is written by a human being.

Machines are able to do computations with any data that we have virtually in the form of 0 or 1. Be it a word file or a excel file, machines can make computations and clumsy calculations over them. But the real problem arises when anything is written on a hard copy be it a diary written by a human being or a book that has been directly published on the person's handwriting itself. Machines can't make any calculations or computations using that type of data, whereas we all know that if those data can be made into use in a right way then most of the unsolved mysteries can be easily solved.

In today's date there are devices to scan a document and read the text over it, but the problem with those devices is that they are not really good in interpreting the human handwriting and can't really interpret the right alphabets most of the times. Fetching a soft copy from the hard copy is one the major problems that the current generation is facing when it comes to technologies. If tomorrow we are able to interpret the human handwriting, then these areas will be flourished to a great extent which will help the human at the end-

1. All ancient documents which are very valuable and are precious can be secured by making a soft copy.
2. Research domain would he very enhanced as the soft copy of all the ancient research documents would be there on internet as a soft copy and can be accessed anytime from anywhere.
3. Many documents are in various languages making it harder for a normal person to read most documents because maybe they are not familiar with the language. But if a device can understand the language and interpret it for the human being then it would be very feasible for them to go through the content.
4. In today's date also all medical documents be it prescriptions or diet plans are written on a paper by a human being, if they can be understood by a machine then it would be very feasible for the humans in getting a proper knowledge of the medicines, they are taking increasing the transparency between patient and the doctor.

MOTIVATION—

Our main motivation behind solving this problem was to help out people who want to enhance this branch of study or want to use the current technology that we have in today's date to normal people. Our main goal is to make a user interface for normal people so that people can upload their document and make the soft copy downloaded on their system. There are many challenges that this technology is facing because CNN is itself very clumsy and needs advanced knowledge of linear regression for coming up with new techniques to leverage this process.

We know that human being has a great need of this technology in today's date, it's an era where we all are living with AI technologies like Alexa where human being can talk with a system can take the help of them in fulfilling the tasks. It not only helps the normal people but also people with disabilities, if they can directly talk and perform tasks through their devices then it is most feasible way for them to experience the emerging technologies. Ai is often looked like an evil in most movies and tv series but people often forget that it is not actually a evil rather it's a friend which would be helping human being in the near future. AI would be the best friend a person can have in their life which they can trust as a machine always performs tasks without making a bias. If this problem of handwriting is solved and the machines are able to read the documents, it would invite many non-technical people towards it.

At that time, they would need a GUI to use the tool so that they can use the feature efficiently. Even if we look from the startup's perspective, if a good GUI would be there for the non-=technical people to understand then many companies would be funding such technologies and this branch of study would be thrived a lot. Our main goal is to make it as user friendly as possible so that people who have no knowledge of systems can also navigate themselves into the tool and leverage the current features that we have in this domain.

In today's date technologies like OCR (Optical Character Recognition) is a mainstream technology in this field. Platforms like Microsoft Azure and Amazon Web Service (AWS) are also having api's to integrate these technologies to any application. Other than this, CNN also plays a major role when it comes to image to text recognition technologies.

We will be using python to solve the issue and modules like pandas, keras and cvs would be getting used while developing the project. Other than these modules we would also be using Tkinter module of python for building the GUI as our goal is to make a user interface of the project which would help the users easy to use and a non-technical person would also be able to use the tool to at least perform the handwriting detection.

1. Handwritten character Recognition using Neural Network:

This project's purpose was to take handwritten characters that are scanned, processed, and a neural network algorithm that is trained with them which would then find patterns, and then change the character . The goal of this research was to create software that would recognise the characters.

2. The automatic conversion of text as it is written on a specific digitizer or PDA, where a sensor picks up pen-tip movements as well as pen-up/pen-down switching, is known as on-line handwriting recognition. Digital ink is a data that can be viewed as a dynamic depiction of handwriting. The signal is converted into letter codes

that can be read by computers and text processors. An online handwriting recognition interface usually consists of the following components: 1) a writing pen or stylus for the user.
2) a touch-sensitive surface that can be combined with or placed next to an output display. 3) a software programme that reads the stylus' movements over the writing surface and converts the strokes into digital text.

3. The technique of automatically transforming text in an image into letter codes that may be used in computer and text-processing programmes is known as offline handwriting recognition. The data on this form is assumed to be a static representation of handwriting. Offline handwriting recognition is more difficult because different people have different handwriting styles. As of today, there is no OCR/ICR engine that allows handwriting recognition.

4. Handwritten Character Recognition : The goal was to identify characters in documents that are scanned and then investigate the impact of modifying the ANN Models. Neural networks are generally used to recognise the patterns . The Neural Network is often used in Optical Character Recognition. The number of Hidden Layers, the size of the Hidden Layers, and the number of epochs were all taken into account. They employed a back propagation Multilayer Feed Forward network. They used several fundamental methods for character segmentation, normalisation, and de-skewing in the preprocessing stage.

5. Recognition of Handwritten Characters Using Gradient Features
Any recognition system must include feature extraction as a component. The purpose of feature extraction was to determine the pattern's characteristics. The gradient depicts the extent and direction of the most significant intensity shift in a small area surrounding each pixel. The Sobel operator is used to calculate gradients. An attempt was made to recognise characters, and the recognition accuracy was found to be 94 percent. Gradient Features were utilised for recognition because of the simplicity of the logic, convenience of use, and high recognition rate.

6. Matlab's Neural Network Toolbox is used to recognise characters.

Recognition of handwritten text has been one of the most active and hard areas of research In the field of image processing and pattern recognition,. It has a wide range of uses, including reading aids for the bank checks, and the conversion of any handwritten document into structured text. Any recognition system must include feature extraction as a component. The purpose of feature extraction was to determine the pattern's characteristics. The gradient depicts the extent and direction of the most significant intensity shift in a small area surrounding each pixel. The first phase is image acquisition, which entails scanning the image, noise filtering, smoothing, and normalisation, and then preparing the image for segmentation, which entails dividing the image into sub images. Feature Extraction increases the rate of recognition and reduces misclassification. To train the neural network to identify and recognise handwritten characters, they employed a character extraction and edge detection approach.

7. Neural based handwritten character recognition
Here the exploration of the existing ring based method (W.I.Reber, 1987), the new sector based method and the combination of these, termed the Fusion method for the recognition of handwritten English capital letters has been done. The variability associated with the characteristics is accounted for in the ring-based approach by considering a fixed number of concentric rings, and in the sector approach by considering a fixed number of sectors. In the training of reference characters and subsequent recognition of test characters, local features such as normalised vector lengths and angles derived from either ring or sector approaches are used, whereas structural features such as end points, junction points, and the number of branches are used in character preclassification. The rate of acknowledgement is encouraging.

8. Character recognition utilising a character geometry-based feature extraction approach.

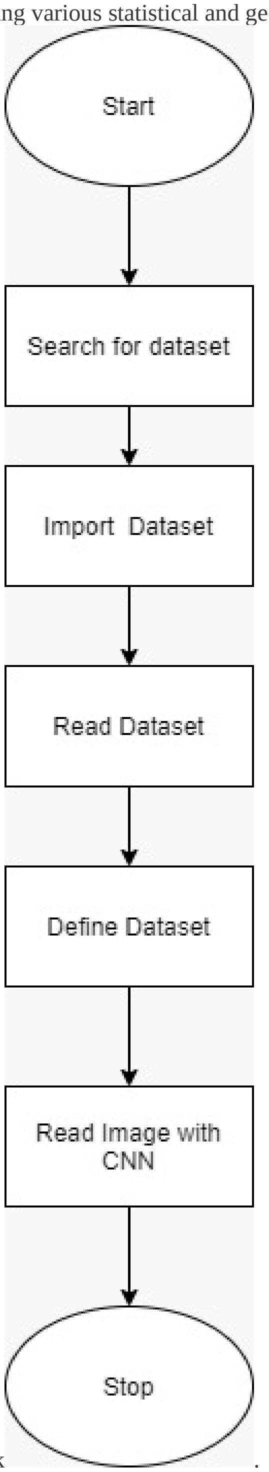
This work describes a feature extraction technique that is based on geometry that can be used in segmentation-based word recognition systems. The suggested system extracts the character contour's geometric features. These properties are based on the character skeleton's basic line types. The system's output is a set of feature vectors. The feature vectors obtained from a training set were then utilised to train a Neural Network-based pattern recognition engine, allowing the system to be benchmarked.

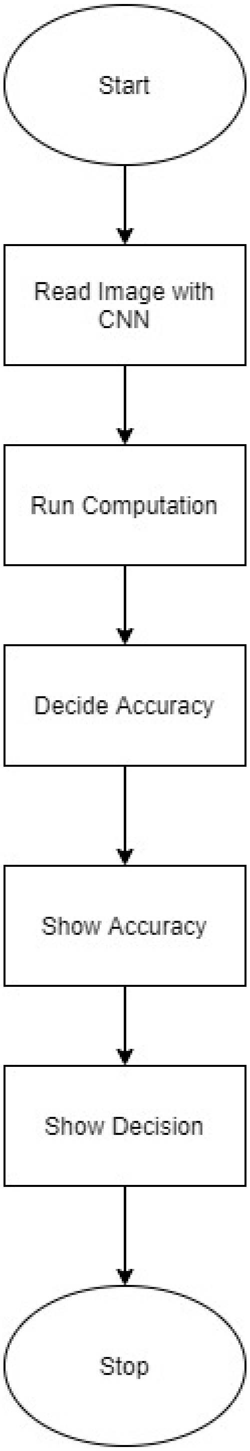
9. For Object Detection, a Review of Gradient-Based and Edge-Based Feature Extraction Methods.

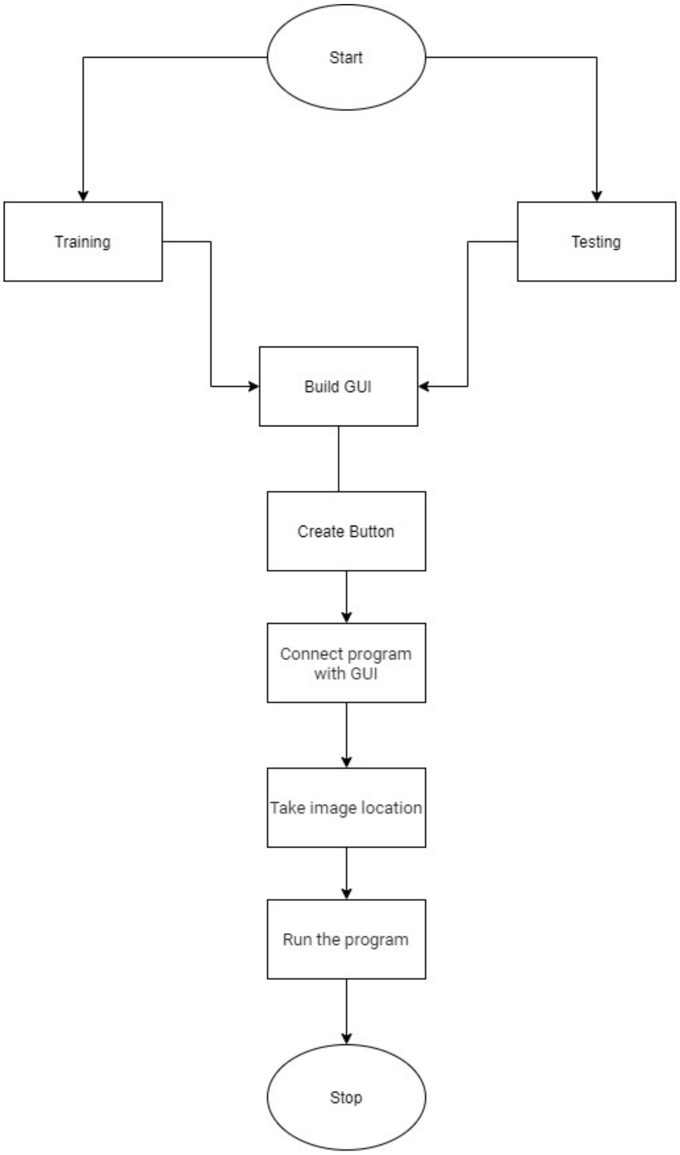
Object detection using image processing is the task of recognising a specific object on a static image or a sequence of video frames in computer vision research. Such research-based programmes have a wide range of industrial and societal applications. These applications can be employed in a variety of industries, including security monitoring, intelligent transportation systems, automated manufacturing, quality control, and supply chain management. Those methods have been thoroughly investigated in a number of research articles, and later studies have proved their usefulness in computer vision research. We separate these methods into gradient-based and edge-based feature extraction methods depending on the low-level characteristics they use. Because a picture can also be viewed as a grid of image patches, the granules concept can be applied to gradient for a review.

10. Many regional languages around the world have various writing styles that HCR systems can identify with the right algorithm and approaches. The existence of odd characters or similar shapes for many characters makes it difficult to recognise handwritten characters, according to research. After the scanned image has been pre-processed to ensure a clean image, the characters are separated into different characters. The preprocessing step includes normalisation and filtration, which results in a noise-free and clear output. Our evolution algorithm will be successful and efficient if we manage it properly with proper training, evaluation, and other step-by-step processes. The recognition of English characters will be

improved by combining various statistical and geometric elements with a neural network .







For handwriting detection system, we would first search for a dataset on internet and use them to train our model after importing them on our system. The training data would be around 70% and 30% would be testing data. After the model is trained with some datasets, we would move to the testing part of the model. Here for testing we need to check the accuracy of the model for which we would be using the pandas module to print a percentage to which the model think the result to be true. This would not only give us a better overview that the model is working good or bad, it would also help us to correct several wrong decisions that the model can make.

Now for the training data, we would first read the image with the python modules. After we read the image on our system, then we would do normalization of the data and give it a shape to make it readable . We would also need to use pipeline for calculating the accuracy of the decisions, we would be using it only on the development part. The main project would not show the users with the accuracy of the decisions because that is not too important for the users as developers. As we would be done with the normalization of the data, we would also need to define features of the dataset because without it the data cannot be read by the model and the training part needs it. As our project also involves images so the use of CNN would be there to make decisions and make the machine understand the handwriting. CNN for image recognitions is provided in one of the famous modules named keras and our idea is to leverage that module to make decisions.

As we would be done with the training part of the model, it's time for the testing of the model and check how accurate the model is working. For testing of the model we would be using sklearn module of python. As the testing would be over it would need to change the image to text and make the decision that which letter is written by the user on the paper. The processing would not be too fast due to the limitations of CNN which are=

1) High computational time
2) Need of high end devices for better computations
3) Need of high end graphic cards to process the images.

As students we would be facing these issues and this is one of the problems that we are not covering on our project. Our idea is to leverage the technology for non-technical people. So, we would move towards the GUI part after our model is ready on development level.

For building a GUI for the users it would be necessary to come up with a design of the application. We would come up with a normal design for the users which would be easy to use for a non-technical people. And they can directly use the technology without worrying about the machines working in the background. The GUI would be build using Tkinter module of python which is the best GUI module at present on python and our idea is to make the interface of maximum 3 pages to minimum 1 page. The interface would not be complicated , we would tell the user that the computations can take time and even can heat up their device so that they can prepare the device as per the instructions and then they can give the location of the file that they want to do the processing over. Of course there would be limitations over the page size as if the page have a too dense hand writing, then it would damage any device and we can't perform such task as a safety measure so if the program would be taking a lot of time then it would be terminated keeping the users security over the mind. This feature is already there on python in today's date that the program stops if it takes a lot of time but as a non=technical person would be using it, it is one of the good features for them. There will be maximum 1-4 buttons on a page including run button to run the program exit button to exit the whole application and some other buttons for instructions and contact .

PROPOSED METHODOLOGY-

As per the project need, we were required for a highly populated dataset to train our model because this branch of detection is majorly based on good testing of the model on a good dataset.

Firstly, we split the data into two halves, one was for the labels and another one for the images. Based on the labels of the data like L we read the whole y data for getting the image. For this thing we are using the split function that is inbuilt csv. After successful splitting of data we reshaped the data based on the training x values to make the data ready to be displayed as an image which we would be giving to the "KERAS" module of python. During this process we shape both our testing and training data on the same constraints itself. Initially in the dataset that we are using we have 784 columns of pixel data which can be converted to a 28x28 pixels. The values on the dataset are in floating point . But the issue is that we can't use floating point values on the main function, we need an integer values so we created a loop to map the integer values with the characters of the dataset.

Process before training the data-

1. Till now we are only explaining the distribution of the alphabets in the dataset.
2. We need to convert the labels into our integer values and store it into a list according to the labels.
3. This list would have the number of images of pixels that are there on the dataset based on each alphabet.
4. Now we needed another list for saving the values of the characters for which we defined a new function into the project.

Now to visualize the data we needed some form of graphical representation. Bar graphs are widely used for visualizing such types of data, because we need to check the number of alphabet and their integer values traffic in short in the dataset. It’s important to know the traffic of a character because no dataset is perfect. If we need to know that why a model is not working fine with some characters in the near future we need to keep an eye on the training part of the model. By knowing the traffic of a character in a dataset we get an idea that the model is weak when it comes to some specific characters because the traffic of those characters is very less. This is very useful in scalability because if a model has not received enough training and testing in a certain areas one can use other datasets for better outputs.

Now we need to shuffle the data to make it ready for training. Shuffling is done direly using the inbuilt function of python. We are not shuffling the testing data as we didn’t had the need to shuffle it.

Keras is now used to change that rough data into an image. The shape of the show function is 3x3 in short there will be 9 plots in one time. Plots are alphabets in this case.

Now we need to reshape the training and testing dataset to move forward with the model. The shape of the training data is made on configuration -

1. [297960,28,28,1]
2. [74490,28,28,1[

Now we needed to create categories because for moving into the testing of the model we need to see the accuracy of the model and how precise the model is showing the results. For this we needed to take a deep dive into the CNN model which takes the labels as the inputs and should give output as a vector of the probabilities. Let’s take a scenario to understand it easier-

Scenario-
Take an instance that the model is being trained successfully and now we need to test the model that how accurately it is working. Now we gave the model some integer values (genuinely we are using keras so these float values would be pixels values and this would form the images in the screen. Taking it hypothetically , let’s say it is “B” , now the model has several sets of alphabets that looks similar to this alphabet. A model is never 100% sure about anything . So, let’s take the model thinks the alphabet as

A	3%
B	30%
C	10%
D	7%
E	20%
F	10%
G	10%

Let’s take the major percentages are like this. CNN is a technology which is still not enough developed in Machine Learning so a model thinking B as A is something that may look weird to a person but for a model it’s completely natural. There will be all 26 alphabets distributed in some percentage , yes some of them can also have 0% value. Now in these type of cases we need to see the other major percentages also, we can’t directly say that the alphabet is B. Though it has maximum percentage. We need to visualize every single percentage chance to get an idea of the behavior of the model.

Reason for this is simple. For now the model is at right track for B but let’s say for C it is predicting as A which is not the case here. In that case how one is supposed to check the problem in the model. Here comes this method of categorizing into use. At every alphabet while production we need to see the percentage change so that we can get an idea that if the model is predicting C as A then which training part are we lacking.

- Are we lacking the training of alphabet C?
- Are we lacking in training of alphabet A?
- Are we lacking in the logic of the model?

Now while training we faced one issue with the dataset. The size of the dataset was around 1GB and training it on one go is not really feasible for our systems. The system configuration that we had while training was i5 8gb ram, which was not enough to meet the requirement. To tackle this issue we used epoch, we used multiple epochs while training our model. It was tiring and time taking at the same time. It took us 48 hours to training the model. But we achieved accuracy in the whole process. The model’s training as done in multiple epochs actually enhanced it’s accuracy marginally and was system friendly also.

Now after the training it’s necessary to take a summarize the data. To achieve that we used summary function which is an inbuilt function for getting a summary of the whole thing including layers and all. The model is saved using save in-built function.

After training it was necessary to test the model on the dataset for which I used predict in-built function of keras which let’s us get a text recognition.

The dataset in a whole performed really well we were able to achieve a good accuracy on the model. There were many instances the model really didn’t performed that well , due to some lack of data. But most of the alphabets performed well during the testing of the model. Modules that we used for the training ,testing etc. are-

1. Keras (For whole CNN)
2. Numpy (For all graphical representations)
3. Pandas (Same role as Numpy but also used for calculations part)
4. Sklearn (Training and testing of model)
5. Matplotlib (Similar purpose to Pandas)

RESULT AND ANALYSIS:_

The F-score, also known as the F1-score, is a metric for how accurate a model is on a given dataset. It's used to assess binary classification systems that divide examples into 'positive' and 'negative' categories. The F-score, which is defined as the harmonic mean of the model's precision and recall, is a technique of combining the model's precision and recall.

The F-score is a widely used metric for evaluating information retrieval systems such as search engines, as well as a variety of machine learning models, especially in natural language processing. It's possible to tweak the F-score such that precision takes precedence over recall, or vice versa. The F0.5-score and the F2-score, as well as the conventional F1-score, are common adjusted F-scores.
F-score Formula Symbols Explained
recall

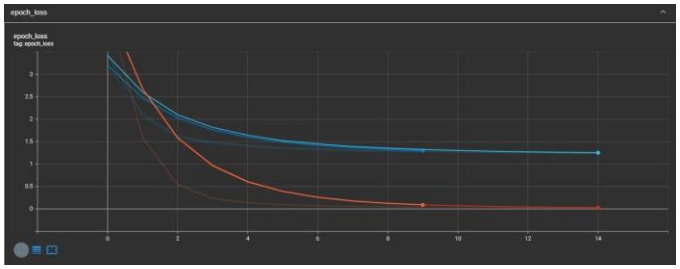
The fraction of positive examples labelled as positive among the total number of positive examples is known as recall, also known as sensitivity. To put it another way, the number of true positives divided by the total number of true positives and false negatives.
precision

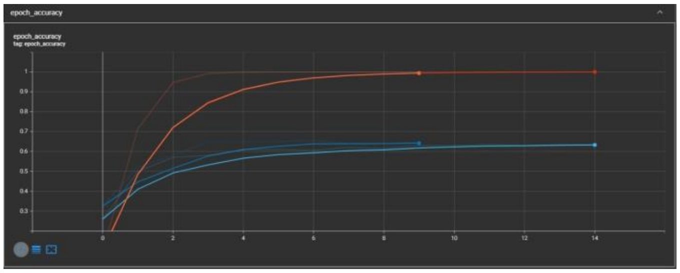
Precision is the percentage of true positive examples among the positive examples classified by the model. To put it another way, the number of true positives is calculated by dividing the number of false positives by the number of true positives + true positives.
A binary classification model can be evaluated using a variety of measures, with accuracy being one of the most straightforward. The number of correctly classified examples divided by the total number of examples is the definition of accuracy. Accuracy is useful, but it ignores the nuances of class imbalances and the different reults of false negatives and false positives.

The F1-score is relevant in the following situations:

- where the costs of false positives and false negatives are different.
- or where there is a significant social divide, such as when 10% of apples on trees are immature. The accuracy would be misleading in this situation, because a classifier that automatically labels all apples as ripe achieves 90% accuracy, but is useless in real-world applications.

The advantage of accuracy is that it is relatively easy to interpret, but the downside is that it is not robust when the data is unevenly distributed or when a particular sort of inaccuracy has a larger cost associated with it.





Roc Curve Sklearn
Receiver Operating Characteristic (ROC)
True positive rate is usually plotted on the Y axis, whereas false positive rate is plotted on the X axis in ROC curves. This suggests that the "ideal" point is in the upper left corner of the figure, with a false positive rate of zero and a true positive rate of one. This isn't entirely accurate, but it does imply that a greater area under the curve (AUC) is usually preferable.
Because it is desired to maximise the true positive rate while lowering the false positive rate, the "steepness" of ROC curves is also crucial.
ROC curves are commonly used in binary classification to investigate a classifier's output.
Binarizing the output is required to expand the ROC curve and ROC area to multi-label classification. Each label can have its own ROC curve, but each element of the label indicator matrix can be treated as a binary prediction to create a ROC curve (micro-averaging).
Macro-averaging, which assigns equal weight to each label's categorization, is another evaluation measure for multi-label classification.
When it comes to machine learning model accuracy is one of the things that we can't ignore and in real world it actually plays a major role in the real world. In the world of AI accuracy is defined in terms of correct prediction of the model. To take a case as an example, let us use our case only for better explanation. Like in our case , accuracy means that how precise and correct the prediction is done by the model. If we are reading A through the model then it should predict it as A only. And when it comes to preciseness the correct prediction in taken into consideration. Our model at present actually works good when the images that we give it to test are similar to the ones on which they are tested on. Genuinely any model in today's date would perform good when we provide them a similar data. The main question here therefore arises that how good it performs when it is given an image that is different from the ones on which it is trained upon. The performance here is actually fluctuating. The reason behind this is genuinely that we have some limitations over CNN. We can't really expect a model to predict every version of A correctly. So, in short it is preforming average on all types of predictions and the accuracy is not too good but it good enough to make a correct prediction.

COMPARISON:-

A similar project was made where tensor flow was used to make a model with trained data, numpy and OpenCV for image processing. In order to suit the project's needs, we developed the software using PYTHON version 3 as the programming language. For the first time, we've employed SPYDER as a tool for research and development. This is Python-friendly and facilitates task completion. We have a number of different libraries and pre-trained models.
Whereas we also used Keras matloplotlib as it acts as the interface for tensorflow and it also has a lot of pre-trained data. Our UI was better but we failed to give the required time for the training and cleaning of the data.
In order to suit the project's needs, we developed the software using PYTHON version 3 as the programming language. For the first time, we've employed SPYDER as a tool for research and development. This is Python-friendly and facilitates task completion. We have a number of different libraries and pre-trained models.
Many papers have been published with research detailing new techniques for the classification of handwritten numerals, characters . The deep belief networks (DBN) with three layers along with a greedy algorithm were investigated for the MNIST dataset and reported an accuracy of 98.75% [25]. The convolutional neural network brings a revolution in the handwriting recognition field and delivered the state-of-the-art performance in this domain [27–32].Features were extracted from a convolutional neural network in their model and achieved 95.96% recognition accuracy.

CONCLUSION:-

In today's world, everyone is part of a generation where machines are capable of performing computations and where machines are leveraging the majority of our possessions.

It doesn't matter if you're making ice cream or creating a document fo r a presentation.
Almost 90% of what a typical human accomplishes in today's world i nvolves robots and artificial intelligence in some way.
During this era, it is critical that machines be made clever enough to r ead our records or ancient sculptures, so that things that humans may not have correctly comprehended or utilised in the best way can be pu t to use with smart machines.

Machines can perform operations and computations on digital dataset and anything fed to them in form of zeros and ones. But the problem is that there are many documents which are not available in that form but they are very valuable and are utilized by many organizations and government as well. One approach would be to convert all those documents into digital data manually which would take ages. So ,our plan is to make handwriting detection system so that we can read that data and convert it quickly and precisely.

There are technologies that can scan a paper and read the text over it in today's world, but the problem with those gadgets is that they are not very good at understanding human handwriting and can't always interpret the correct alphabets. One of the greatest technological challenges that the present generation faces is obtaining a soft copy from a hard copy. If we are able to decipher human handwriting tomorrow, these fields will flourish to a significant level, ultimately benefiting humans.

1. 1. All priceless and valuable antique papers can be safeguarded by making a soft copy.
2. 2. The research domain would be much increased since a soft copy of all historical research documents would be available on the internet as a soft copy that could be viewed at any time and from any location.
3. 3. Many documents are written in different languages, making it difficult for a layperson to comprehend them because they may not be familiar with the language. However, if a technology can understand the language and translate it for a human, it will be much easier for them to read the content.
4. In today's world, all medical documents, such as prescriptions and diet plans, are written on paper by a human being; if they can be understood by a machine, it will be much easier for humans to gain a thorough understanding of the medicines; as a result, they are increasing the transparency between the patient and the doctor.

Our main goal for resolving this issue was to assist people who wish to advance in this field of study or who want to make use of the current technologies available to ordinary people in today's world. Our main goal is to create a user interface that allows ordinary people to submit documents and download soft copies on their computers. There are numerous obstacles that this technology faces, as CNN is inherently clunky and requires sophisticated knowledge of linear regression in order to develop new approaches to utilize this process.
We all know that human beings have a huge need for this technology in today's world; we're living in an era where AI technologies like Alexa allow people to talk to machines and have them assist them in completing chores. It benefits not just normal people, but also persons with disabilities; if they can converse and conduct things directly through their devices, it is the most practical way for them to experience modern technologies. In most movies and television shows, AI is depicted as a villain, yet many people forget that it is actually a friend that will be assisting humans in the near future. AI would be the finest buddy a person could have in their life because a machine can always accomplish duties without bias. If the problem of handwriting is overcome and machines can read documents, it will attract a large number of non-technical individuals.
They would require a GUI to use the tool at that time in order to make efficient use of the capability. Even if we look at it from the standpoint of a startup, if a strong GUI is available for non-technical people to grasp, many companies will invest such technologies, and this field of study will thrive greatly. Our main goal is to make it as user-friendly as possible, so that even persons with no prior experience with systems can navigate inside the tool and take advantage of the current features available in this domain.
In today's date technologies like OCR (Optical Character Recognition) is one of the mainstream technologies in this field. Platforms like Microsoft Azure and Amazon Web Service (AWS) are also having api's to integrate these technologies to any application. Other than this, CNN also plays a major role when it comes to image to text recognition technologies. Other than this all some other services in this field are-

- MicroBlink
- Microsoft Azure Read API
- Mitek
- Google Cloud Vision API
- MyScript
- Selvasai
- Unitek.ai
- Vidado
- Hanvon Technology
- Hanwang Technology
- Infrrd.ai

APPROACH AND PERFORMED TASKS:-

We will be using python to solve the issue and modules like pandas, keras and cvs would be getting used while developing the project. Other than these modules we would also be using Tkinter module of python for building the GUI as our goal is to make a user interface of the project which would help the users easy to use and a non-technical person would also be able to use the tool to at least perform the handwriting detection.

For a handwriting identification system, we would first look for a dataset on the internet and then import it into our system to train our model. Around 70% of the data would be for training, while 30% would be for testing. We'd move on to the testing component of the model after the model has been trained with certain datasets. We need to test the model's correctness, therefore we'll use the pandas module to display a percentage of the time that the model believes the answer is correct. This would not only provide us with a better understanding of whether the model is performing well or poorly, but it would also assist us in correcting many errors that the model may make.

For the training data, we'd first use the python modules to read the image. We would normalise the data and give it a shape to make it legible after reading the image on our system. We'd also need to employ pipeline to calculate the correctness of our decisions; however, we'd only use it for development. The accuracy of the decisions would not be shown to the users in the primary project because it is not significant to the users as developers. After we've completed the data normalisation, we'll need to define dataset features because the data cannot be read by the model without them, and the training section requires them. Because our project includes photos, we'll utilise CNN to make judgments and teach the machine how to read handwriting.

One of the well-known modules, keras, provides CNN for image recognition, and our plan is to use that module to make decisions.

As we would be done with the training part of the model, it’s time for the testing of the model and check how accurate the model is working. For testing of the model we would be using sklearn module of python. As the testing would be over it would need to change the image to text and make the decision that which letter is written by the user on the paper. The processing would not be too fast due to the limitations of CNN which are-

1) High computational time
2) Need of high end devices for better computations
3) Need of high end graphic cards to process the images.

As students, we would be confronted with these concerns, and this is one of the issues that our initiative does not address. Our goal is to make technology accessible to non-technical people. So, after our model is ready at the development level, we'll move on to the GUI phase.

It would be necessary to create a design for the application in order to establish a user interface. We would provide a standard design for the consumers that would be simple to utilise for non-technical individuals. They can also use the technology without having to worry about the devices in the background. The GUI would be created with Python's Tkinter module, which is currently the best GUI module available, and our goal is to reduce the interface from a maximum of three pages to a minimum of one page. The user interface would not be complicated; we would inform them that computations can take time and even heat up their device, allowing them to prepare their device according to the instructions before providing the location of the file they wish to execute.

Of course, there would be constraints on the page size because if the page had too dense handwriting, it would harm any device and we wouldn't be able to complete such a task as a safety measure, so if the programme took a long time, it would be terminated with the users' safety in mind. This feature is already present in Python today, in that the programme will halt if it takes a long time, but it is one of the useful aspects for non-technical users. On each page, there will be no more than 1-4 buttons, including a run button that will execute the programme, an exit button that will exit the entire application, and some other buttons for instructions and contact.

REFERENCES:-

➢ Chirag I Patel,Ripal Patel, Palak Patel.
https://www.ijser.org/viewPaperDetail.aspx?MAY1114

- S. Basavaraj Patil, N. V. Subbareddy ‘Neural network based system for script identification in Indian documents’ in Sadhana Vol. 27, Part 1,February 2002, pp.83–97.
- T. V. Ashwin, P. S. Sastry ’A font and size-independent OCR system for printed Kannada documents using support vector machines’ in Sadhana Vol. 27, Part 1, February 2002, pp. 35–58.
- Kavallieratou, E.; Fakotakis, N.; Kokkinakis, G.,’ New algorithms for skewing correction and slant removal on word-level [OCR]’ in Proceedings of ICECS '99.
- Simmon Tanner, “Deciding whether Optical Character Recognition is Feasible”.
- Matthew Ziegler, “Handwritten Numeral Recognition via Neural Networks with Novel Preprocessing Schemes”.

➢ AshutoshAggarwal, Rajneesh Rani, RenuDhir
https://www.researchgate.net/publication/280063550_Handwritten_Devanagari_Character_Recognition_Using_Gradient_Features

➢ Kauleshwar Prasad, Devvrat C. Nigam, AshmikaLakhotiya and DheerenUmre.
https://www.researchgate.net/publication/317174185_Enhancement_of_ANN-Based_Offline_Hand_Written_Character_Recognition_Using_Gradient_and_Geometric_Feature_Extraction_Techniques

Hanmandlu M, Murali Mohan K.R,Kumar H.
Dinesh Dileep

➢ Sheng Wang.
https://www.researchgate.net/publication/355711639_Handwritten_Font_Image_Design_System_Based_on_Deep_Learning_Algorithm

➢ Hitesh Mohapatra (KL University)
(PDF) Handwritten Character Recognition (HCR) Using Neural Network (researchgate.net)

➢ J.Pradeep , E.Srinivasan and S.Himavathi
https://arxiv.org/ftp/arxiv/papers/1103/1103.0365.pdf