Final Project  
OCR

Optical Character Recognition Application

P16190188

Joshua Mbanugo

Abstract

This report is about the stages of my development and research throughout this project I am creating. This project is OCR application that allows the user to input data into the app by drawing it using a touch screen interface and then producing an output of the digital version of the symbol the user had created.

My final project will be involving the use of OCR which is a character recognition application. I have developing this application for android so that it will be mobile to use. The development stages have been through a lot of stages of research on what data type I would need to use as well as what libraries I would need to add or include into my code. OCR have been known to not be that great bringing out false outputs. The main interest in this project is the use of AI since that is a well developing new piece of technology that is still being updated and used in current day. The use of AI will be used in the future and devolved with more advanced features but as for today I needed to understand what it is that makes an AI a lot more special than normal commands or script that has created to do the same thing.  
  
An AI is a consistent set of rules or connections through a neural network create relations and dissections based on what the input is. This makes it much more advance and unique with the property of machine learning. The main part of me machine learning is to make decisions based on its own observations making it critical for the program to train before running.

With developing OCR first challenge I came across was what language and what programs support OCR and what software would be better for me to create it on my android device. I investigated some languages that support Optical recognition and came across a few but the main ones I saw was Java, C and python. Since Java was the main language I wanted to learn in I focused on software the support java and that’s when I came across android studio. Android studio is a development workspace that allows a user to code in java and C. the great thing about it is that it also allows other languages for designing the front end of the application in languages like xml which is supported by android devices.

This also allowed me to expand my knowledge on other languages but to be able to put two languages together into one application, which was something I hadn’t come across and interested in. The other feature I found out that I needed to understand what exactly I needed to implement on to my project. That included the libraries, data sets, script changes like “build.gradle”. these were important to get understand so that I would understand how to program it when it got to the code.

When I was looking at other programs I found out the fuzzy can be used in creating a neural network, fuzzy is well known to be used for its rules and connections between each membership so I investigated that to find out if fuzzy was a better approach.

Fuzzy logic is a method used in many the uses rules. It’s one of the few that has great ways to solve and get absolute values with logical systems and fuzzy sets. Fuzzy logic has greater and specific values that goes through Boolean values but unlike traditional logic, fuzzy logic scales within the values of 1 and 0. For traditional logic to be true it will have to be 1 or 0 specifically but with fuzzy logic you can have a statement to be true but with a value of 0.7. the reason why this is a useful methodology is because when you a looking for a value with u need between 2 inputs and you want to know if it’s true of false. These two inputs when they are both relevant won’t be (most likely) be on the same value of 0 or 1 so you would want to know in-between.

Fuzzy logic has been around and been used in a lot of AI advanced technology to get the values of inputs. It takes the inputs and find were on the graph the points line up, with this it gets the values and processes them through I list of rules were each value is relevant to create a final output. The fuzzy set is then processed into a crisp set for that corresponds to a membership degree. This gets a fixed value were the member ship degree and with the information gathered we can determine what procedure would need to be taken place for example sunny turn the lights off but increase the light dimness based on how bight it is outside.

Optical Character recognition is used in some parts of fuzzy logic with the use of facilitates design of systems. This helps it mimic humans reasonings through neural networks on how many neural connections it has with its data it needs to identify. For OCR to work the system needs to identify optical characters and recognise them as specific letters. They do this by laying out the convers of the letters and using measurable algorithms through pixels to identify what character it is based on its Hight and length. The basic canvas size of a letter is 64x1 pixels. This is a sensor the processes the data input and formats it into a more fuzziable approach.

With the results of the letter/numbers that has been inputted it is then given a set of rules based on how wide it is and how high it is. And then sent to its fuzzy membership. For it to get to its physical form the modification of the fuzzy membership functions and set rules would need to be changed.

Since a lot of the number out there has ruefully the same width and Hight would have to set some rules were both the letter would apply but not on the other and based on how much one rules fuses with the other it will determine how much of that number it is. The set of the 0 and 8 has a specified width of 91 mils. The other number of 4, 6 and 9 has a width of 78, the numbers of 3,5,7 is specified at 65 and the final numbers of 1 and 2 has a mil of 52 mils.

Fuzzy logic has been optical recognition before with identifying license plates with ease. The great thing about these are that they are quiet to recognise since all it takes is a picture.

My aim

The main target that I want to accomplish is to create an application that will allow a user to be able to interact with my application by drawing a symbol, which will digitalise it and recognise the letter they have drawn. With this Idea I had to come across the main features on what I needed to do to make this project maintainable as well as have a structured development that I could work with. First, I needed to find out what development approach I would like to have. Based on how I work and how the project is going to be, I chose at the end to have an agile approach. The reason for this is because the application I am making will have a lot of room of upgrading and development in the future and when I’m making the app I would like to add some features I didn’t tend to add in the plan that could make the project better. This would and could create a positive or negative effect on the project but since its an agile approach I still will be able to go back to fix what is broken.

Reequipments

The reequipments of my project was the next thing I needed to identify since it was a key part of development. This would create many open doors of improvements as well as tell me a lot of the risk that I could stumble across. For my project I needed the understanding of a neural network and all the possible types of data and engines I could read before thinking about developing the project. The first title I came across was machine learning. With machine learning there is a lot of familiar aspects that I have already read about such as neural networks and node collection. These were important for my project, so I investigated the futures that I needed to implement into my OCR.

Another aspect I needed to investigate was what device is this going to be on. I needed to know this because of the language I would be coding in because some languages are better for some platforms then others. Plus, I wanted a very open operating system to work with that had very little limitations and have a device to test it on. So, with all this I chose it will be an android device. An OCR would be great on my phone because it would be able to test as well as to code in the language of java. This was the main language is choosing to code my application in. The reasons for this are because I am very familiar in the language but still want to develop my skills in getting bet in java. Also, for testing I only have access to an android phone and I could have used an iPhone emulator, but I chose not to because I would like to see my application of a fully functional phone.

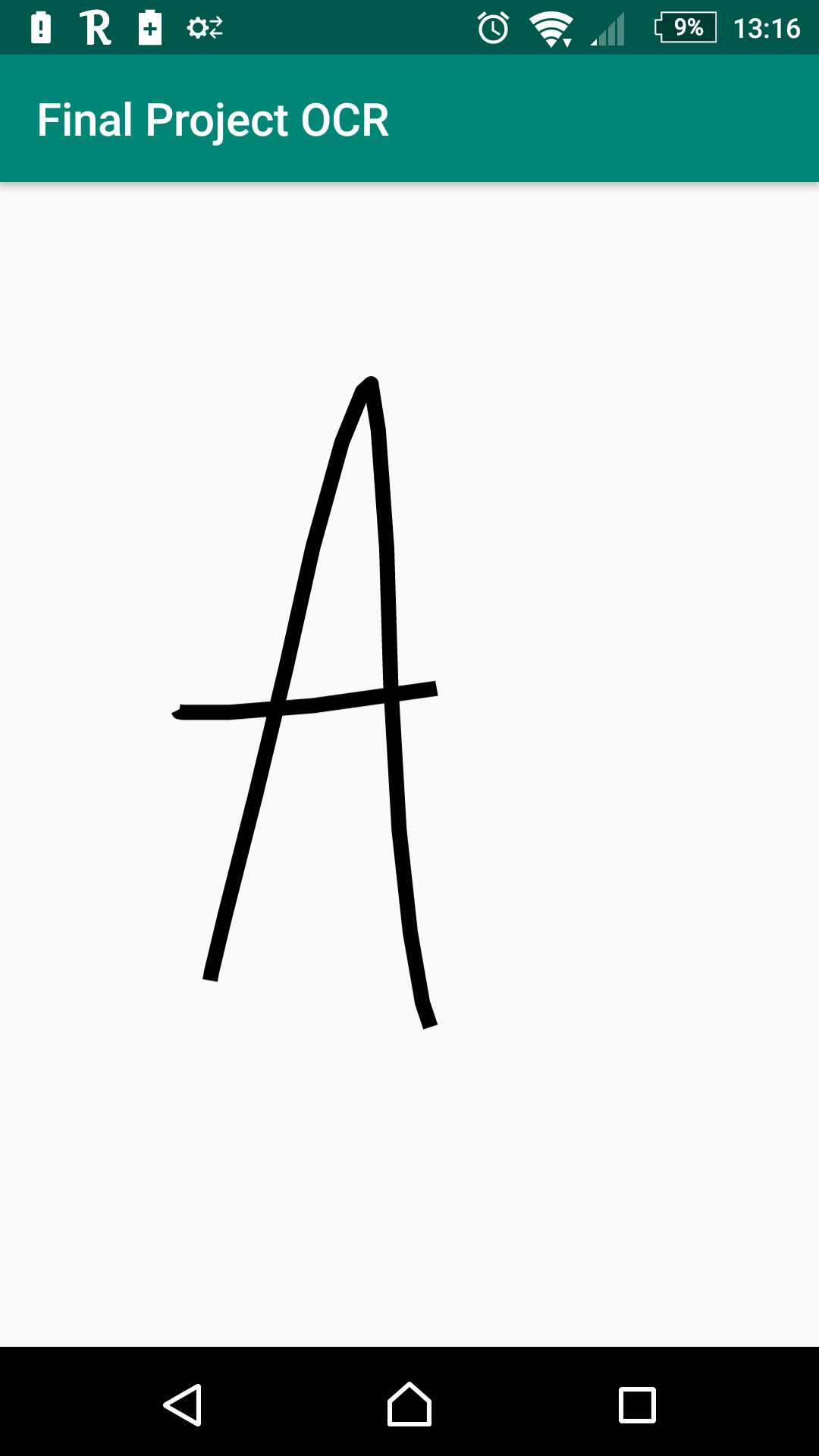
The next step I investigated is the layout of the app in how the application is going t function. The first idea was so that they would only draw the letter and send the output but that lead to many limitations that would be needed to be added. One of the implementations that was added after the current idea was the button to reset and configure and save. Originally the plan was so that the user just drew the symbol and then the app would recognise it and instantly change the page buy telling you what letter it found. The problem with this is that if u want to write the letter M the app could think you are writing “n” and instantly change before you even finish. This issue could be fixed buy adding a timer between each time it had recognised a letter by 2 seconds allowing the user to fully write the letter, but this issue is, if the compiler was confused between two letters it would continuously keep switching the two creating in infinite loop without a button of reset. This would mean that the user would need to close the application to reopen it again to operate the application.

Whilst I was about to finish the reequipments I chose the IDE of android studio with other applications that allowed me to keep track of backups called GitKraken. The first issue that I came across was that I had to learn xml for my application because that was how the IDE turns it into and application. Since xml is like HTML I didn’t have much trouble adapting what I needed to do. GitKraken is like GitHub and has the same functionality. The only difference is that its on your desktop and it automatically creates updates on each save.

Development

In the development there is two parts that I have separated it into.

Testing



This is a small example of what the application is going to look like.

In development with my OCR I had to make some changes with my original model because I forgot about the main functionalities that would e needed on the applications for example the reset canvas and the find and save. These didn’t come into mind because I wanted it to recognise as soon as the letter has been found. The problem of this would be that the user wouldn’t be able to confirm its letter before it will recognise it making it when you draw out the letter A, it will start searching the letter n because it would think the user is done after they have released the finder from the canvas.

I then changed the whole development and drew out what I wanted to have on the application and went through the reequipments that I needed to implement to make sure that the application was doing the right thing. First, I looked in the layout. The layout was one of the biggest hold backs of this development because it would end up to a point the whole code would break because of the compatibility of the version of android. When I was running this app on my new phone I released there was a compatible issue with the version of android I was coding in making it not compatible with the new versions of android. When I noticed this, I found that it was a big issue since I would want this application to be updated as it goes on so that everyone with an android will be able to operate the app in ease.

I then made a different version of the application that supported my new phone which was an android model of a 8.0/8.1. there is a android 9.0 coming out soon so I’m ready to test the app on that version so that there isn’t any issues. When I found out of the issues I then decided to create a different version completely as a backup to work on and call it OCR2. This was so if I made any bugs on their way I would have a version before although it wasn’t up to date the code was able to work for that version, so I would be able to a flex the code to fit the current version.

With the new version I then proceeded in to adding the implementations that I was creating a draft for so that I could have a solid model to work with. The way I planned out my development was with a gentchart that I designed to follow so that I would be up to date as far as if all went smoothly. In some parts I have fallen behind in and tried to recover through different aspects of the application for example. The other day I came with a configure issue with the buttons not identifying that I have drawn something on the canvas, so when pressing the button it would do anything since it thinks nothing is drawn. Another thing that had popped up is that for the program to identify that I have drawn something, it needs a solid background colour which I have chosen to be white since white and black is quite easy to see but since I didn’t manually change it because it looked white, the program seemed to not like it so it didn’t respond.

Creation

Literature review.

OCR stand for optical character recognition this is used to identify symbols by importing its physical form through an algorithm to digitally translate it into an optical form. With OCRs getting popular there are much more problems that come across when creating this technology. For the OCR to recognise a letter of number they would go through a series of possessing that would recognise what character it is. first the main issue they found out when researching this technology is that some OCRs had issues identifying the differences between the letter 0 and o. the program would think it’s the same character and process it as a 0 as it runs through numbers first. Another issue that can affect OCR is the colour background of the text. Therefore, the standard licence plate must be black and white, or black and orange in some places. These colours are easier to see then a white and grey contrast. The easier the colour difference between the images the better it is for the system to recognise the characters. This also give me a question why not use a green and black contrast as the black text would work well if the green would work like a green screen to blank out the background.

With the tech evolving they started branching out using Artificial Neural Network (ANN). This was a great success at the time making the recognition increase by 82%. Then “Pal et al” used support Vector Machines (SVM) which hit a 99.18% accuracy for Devngari and 98.86% accuracy for Bangla characters. The main method that has stood out in all of them is the binarization method for document images by using thresholding method and dynamic windows. This allows to recognise hand written words that a user or person has drawn. In nowadays products there are a lot of OCR software that support desktop, server and web OCR which most of the have and accuracy rate going between 71% to 98%.

In the development stage of OCR they created a open source OCR engine that was created in 1984 to 1994. This was to increase it accuracy of the recognising images and symbols at its time. In late 2005 they released Tesseract for open source. This engine can be used and creating the command prompt which will recognise documents. This can be used in a lot of languages like java and PHP so the accessibility of this engine is high. This works in steps; the first stage is adaptive thresholding which converts the image into a binary images. This is then analysed through connected components which then is extracted to character outlines.

OCR is seen as huge part of the community of technology since there is a lot of room of improvement. It was only recently that Google managed to implement one of there OCR technology into google maps so that it would show them the direction that were they are going through the camera. The main parts of an OCR are the input device and the output. In optical character recognition there is a lot of uses and places you can implement the technology to such as laptop and mobile touch screens. The concern with this is the touch of the screen. With a touch screen you would be able to simply touch the screen but with a laptop you would either need to install some sort of hardware like a graphics tablet or images that has been taken.

The interface is the main part of the OCR that pulls everything together. With each interface there are sensitivities based on the touch of the screen on or on a remote control. Touch devices are the main target in now a day’s technology since we are moving to easily accessible objects. With each PDA (personal digital assistant) everyone owns there will be an evolution on what needs to be added to make life easier for people. The first main thing that is linked to this I the compatibility of other exemplary devices. Exemplary devices are ‘embodiments of electronic devices’ [2] ‘user interfaces for devices’ [2] and processes that are used in devices to operate. Its known to say that exemplary embodiments are implemented into devices like ipad, iphones, and all apple and android products that have the option of touch screen. It is also important to understand some embodiments that are not portable can be compatible with this tech like TVs and desktop computers. All they would need to do is implement a sort of hardware that allows it to become compatible with what the touch screen devices can do.

Recommendation

Things to improve

Conclusion