**PROJECT: ID CARDS FRAUD DETECTION**

**TEAM: SYNACK**

**#LE TRAN TRUNG DAI - LEADER**

**#MAI TIEN NHAT - MOBILE DEVELOPER**

**#NGUYEN CHI THANH - DATA SCIENTIST**

**#LE THANH TIEM - DATA SCIENTIST**

**#LAM TUAN TU - PRODUCT MANAGER**

Ho Chi Minh, Dec 09 2019

1. **Project Overview**

* Context: We are in a competition hosted by Facebook Developers Circle with more than 50 other teams to developing mobile applications that using React Native and Data Science to tackle a real-world problem from one of five companies: MoMo, FE Credit, SenDo, Haravan and ChoTot.
* In this Project, we followed the problem of FE Credit - ID Cards Fraud Detection

1. **Main Problems**

* Scammers are using fake ID cards or others ID card to apply for loans. We have to build some functions to detect the input ID cards are valid or not.
* Perform manual check for a large amount of ID Cards cost a lot of time and efforts and may cause critical mistakes. We have to automate the ID checking process with our application to get the informations from ID cards without human intervention.

1. **Scope of works**

* This project only gave us about 6 weeks – which is pretty short - to build a fully funtional application for the contest. In this case, we decided to follow Lean Startup Sofware Development Model instead of other models like Waterfall, Agile or V-Model to minimize the time gaps between phases and the managerment effort we put in.
* This model had three main steps: Build, Measure and Learn connected as a loop circle and use MVP (Minimum Viable Product) as a core concept for all the development process.
* The project is composed of multiple phases as follows:

• Phase 1 – Documents and Solutions Research (1 week)

• Phase 2 – Make a first MVP (2 weeks)

• Phase 3 – Inproving solutions and developing advanced funtions. (2 weeks)

• Phase 4 – Packed the software and make it ready to race. (1 week)

* The main concept of the project includes construction work associated with Phase 2, 3 and 4, which includes installation and testing of both server side and client side.
* We gonna do all three fraud detection methods the Mentors provided us:

• **Template matching**

• **Smart ORC**

• **Face recognization**.

* We also have additional feature outside these scopes that we will describe later on this document.

1. **Technical Solutions**
2. Template Matching

We developed a feature that can detect any small change on the static input ID cards compare with the template we selected before.





This feature worked well in testing enviroment but maybe useless in real life condition since real ID cards may have many noise or dirt that can trigger the REJECT flag. In case we lower the aceptable threshold of this function, it became too easy to ignore important differences and ofcourse it is not good at all.

After going through a bunch of test cases, we realized that it’s really really hard to choose the best balanced threshold value for Template Matching funtion so we decided to use it another way – as an Object Detector – to detect if the input picture contains the ID card or not and then crop it out of the whole picture. It’s very useful for create the best input data for other functions like OCR or Face Recognition.

1. Smart OCR

We call it “Smart OCR” instead of just OCR because we didn’t just throw the whole image to the OCR Model to read it but we’ve applied many pre-processing methods that auto-tranform and auto-clean the input images that users took from their camera.



Then we sliced it into small pieces before applied our unique color-processing functions to remove the background pattern and noises before convert it to binary images.



After all we use the Tesseract OCR Model that trained with Vietnamese characters in order to get the best results.

We tested with more than a hundred cases. The data chart below shows the accuracy of text data returned on each datafield of ID card after OCR applied.

We also filter the fake case base on OCR result for ID card number and birthday becase ID number must have 9 or 12 number and the owner must older than 16.

1. Face Recognition

We used the template matching method to crop out the face picture on the ID card and compare it with the selfie picture from user’s camera in order to know if the applicant is the real owner of this ID card or not.

To extract the faces, we used a deeplearning method called MTCNN, then we used a pre-trained Keras VGG\_face model to compare it and return the match percentage.

Ofcourse it can be cheated if the user is not taking the pictue of their face but using a printed image instead. In that case, we planned to use a face angle detection to get the movement of the face to decided it’s a real face or not. But anyway, it still can be cheated because you can easily get a selfie video of a persion especially a girl on her facebook stories or instagram. Even the most famous and strongest face recognition module that we all know – the Face ID on iPhone X series – still have weaknesses and can be cheated by a twin or a 3D printed mask. So how can we solve it? How can we tell the exactly owner of an ID card? This question pushed us to find out a better way to authenticate the owner of the ID card.

1. Out of scope feature: Fingerprint recognition

This function is only available for the offline situation when the applicant has a face to face coversation with the counselor. The idea of this method is if you can extract the fingerprint images on the ID card, and get the fingerprint of the applicants, you can 100% confirm the real owner.

We still used the template matching method to crop out the fingerprint picture on the backside of the ID card and compare it with the applicant’s fingerprint. How can we get the applicant’s fingerprint? We used this fingerscanner module. It has a light sensor that scan your fingerprint and convert it into a bitmap image. At this time, we’ve completed the fingerprint comparing function, but unfoturnaterly, at this time, we still have trouble when integrated this module to our application so it can not take the fingerprint and load it to the app in realtime but we still can demo it by upload finger print image we stored in the phone’s gallery.

1. Client and server model
2. **Improvements and future plans**

This is just a prototype of what we planned to do. So, we still have a bunch of things to fix, to improve and to develop in the future. Let’s take a look:

* Improve the template matching functing by collecting and training the model on a real big dataset so it can detect not only this type of ID cards but any other types and maybe other things like driver license or student card… ect. We have a whole world to detect.
* Apply machine learning to the image pre-processing function, to adjust the saturation, brightness and constract. It’s the key to get better OCR result.
* Develop face angle detection and screen detection to prevent scammers to use static images or screen to display other’s picture.
* Complete the fingerprint recognition feature.
* ID card verification and OCR is the turn-key solution not only for this project but also have many applications on other case like national competitons, professional certificate tests, or any business or operation that have to handle a large amount of ID Cards everyday.

1. **Credit**

This project only gave us about 6 weeks – which is pretty short - to build a fully funtional application and also required many more different techniques from what we’ve learn through this course. So even what we wanted to do is very very bigger than what we actually can do we still never give up. Thanks Facebook for this meaningful contest, thanks CoderSchool for all the lessions you teach us, thanks our lovely mentors from FE Credit and persionally I want to give a big thank to my teamates, you are wonderful partners. Every sleepless night together gave us more knowledge and also more unforgetable memories. Thank you all for your attention, this is SYNACK and we’ll see you later.