

FLY THROUGH FACES



200654 Muhammad Ehtasham Qadir
201900 Muhammad Hassan Mukhtar
201982 Sukaina Raza

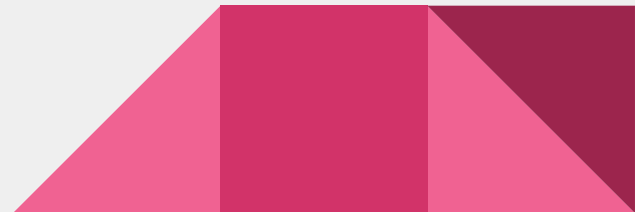
Outline

- Abstract
- FYP Requirements

Why? Need


How? Approach

- Deployment
- System Interaction Design Usage Guideline
- User Testing
- Improvements



Abstract

This project will create a **Software System** to verify a person's identity using their e-passport and a camera. The system captures an image with a camera, and perform verification of owner through **Facial Recognition**. It involves developing software for image capture and processing, and implementing **Deep Learning Algorithms**.



FYP Requirements :: Why?

Paper Passport

Takes a lot of time and resources for **Creation**.

People don't use them full before they expire and paper is **Wasted**.

On Verification

People can dodge/bribe normal counter but not Computer system.

Suppose you have coronavirus the person
may allow you system will not.



FYP Requirements :: How? Implementation

Face Detection

Feature Extraction

Face Recognition

Software

Passport Creation and Verification

FYP Requirements :: How? Requirements

Camera (Video Feed) and **Clear Face View**

Computer or Embedded System (CPU,Memory, and Storage)

Display (Recognized Face or Information related to the Identification)

Face Detection (mediapipe,mtcnn)

Feature Extraction (xception_model, Statistical Pattern Histograms)

Face Recognition (Sigmoid, Euclidean Distance 3D)

Deployment :: System

Desktop Based System

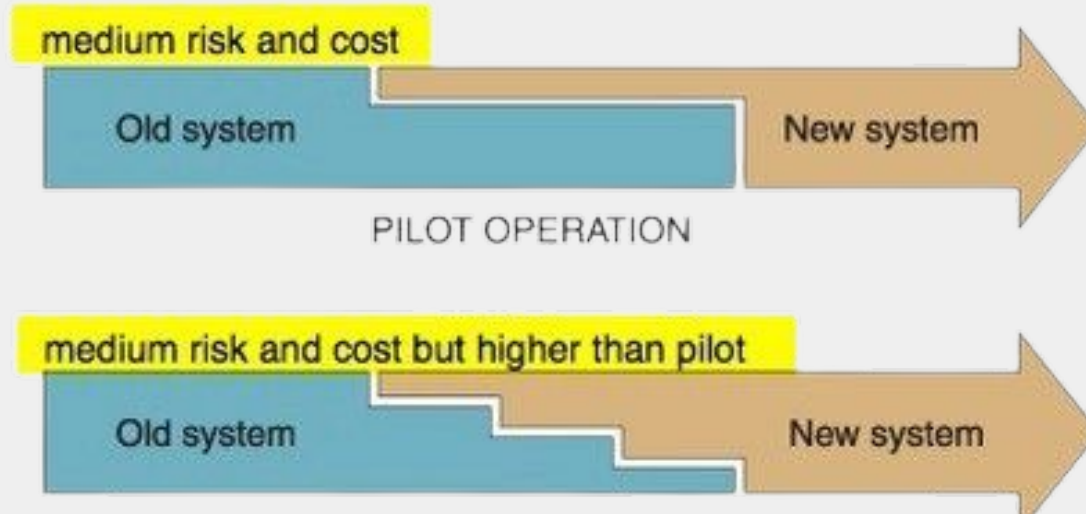
Platform :: Windows

Hardware :: Camera

Software :: All software dependencies are inside .zip file



Deployment :: Procedure

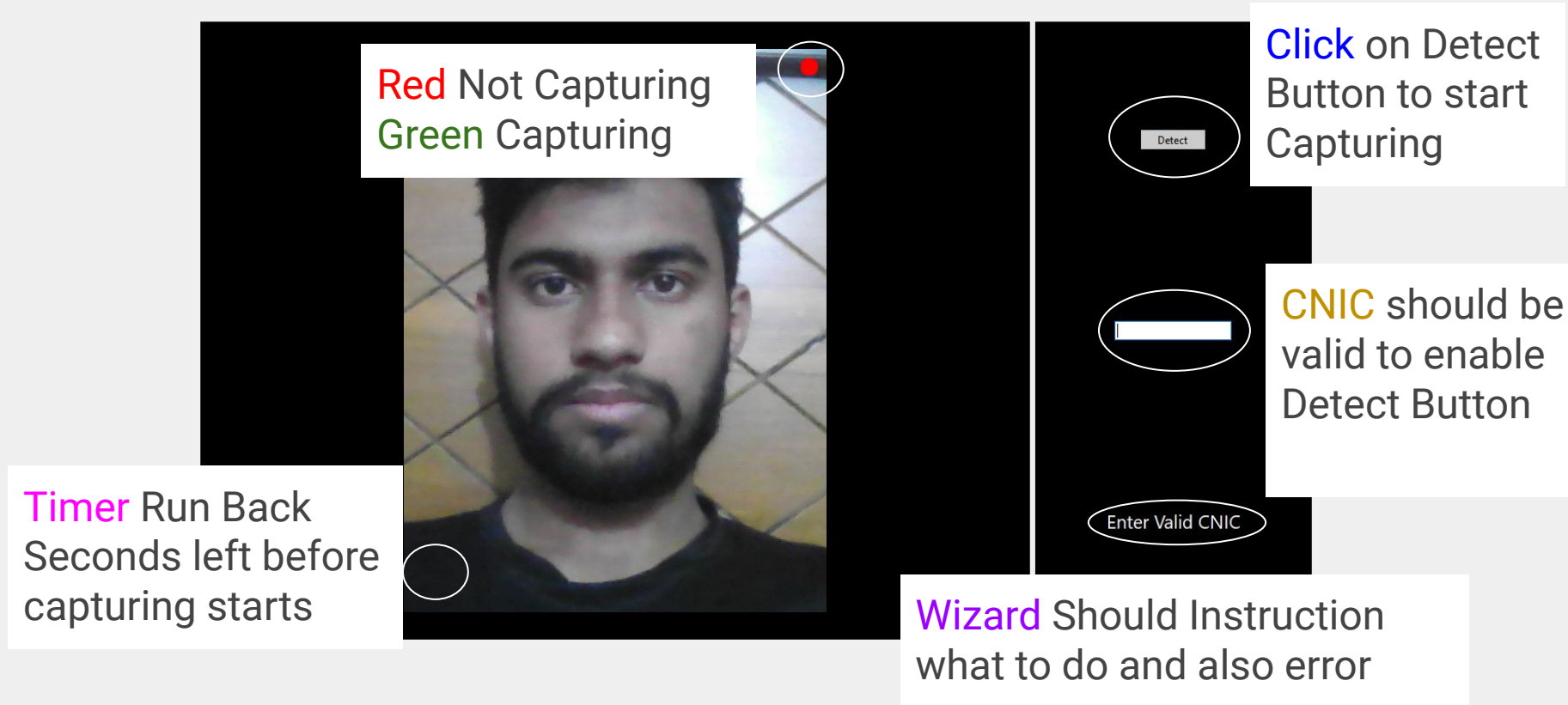


The pilot keeps the old passport verification and adds a new counter for e-passport checks.

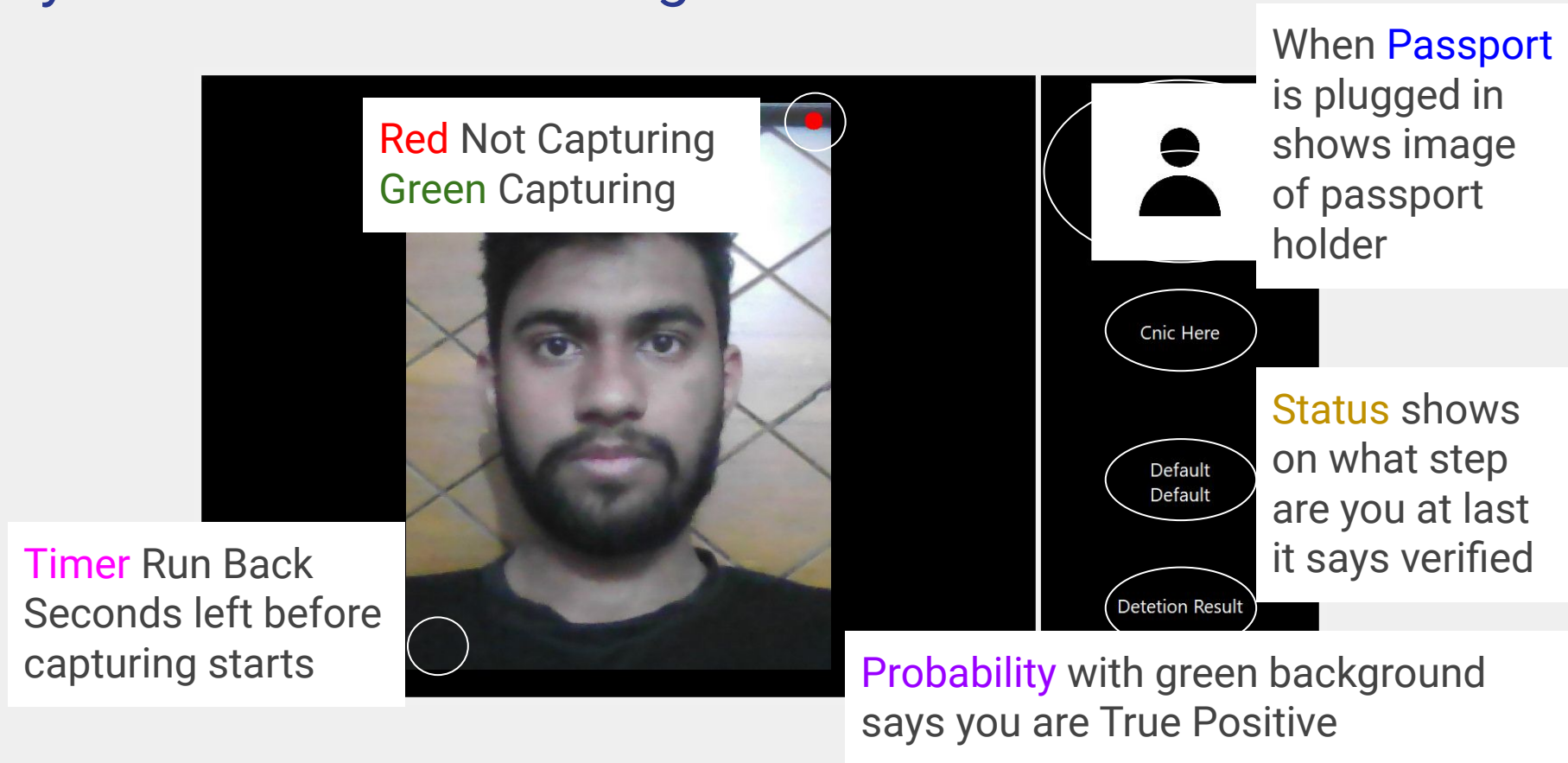
System Interaction Design



System Interaction Design :: Creation



System Interaction Design :: Verification



System Interaction Design

Passport Creation

Enter Valid CNIC

Plug USB Passport IN and Click Detect **User Facing Camera**

Directly Face Camera Above Screen **Wait 3 Sec**

Soon the Screen Shows Passport Creation **Successful**

This passport creation will be done by officer only no user can directly interact with machine.

System Interaction Design

Person Verification

Plug USB Passport IN **User Facing Camera**

Directly Face Camera Above Screen **Wait 15 + 15 Sec**

After 15 seconds, images will be captured. Please remain still for an additional 15 seconds to ensure clear data collection for verification purposes.

Soon the Screen Shows Passport Verification **Successful**

Note :: If **Failed** Try Again Else visit the nearby Helping Staff.

User Testing


Test involved multiple users; 4 out of 5 users were successfully verified.

Verification process determines if passport matches the user's identity or if someone else tries to use it.

Disclaimer this system cannot verify twins 😊

We cannot share the user testing data due to privacy policy restrictions.

**However, you're welcome to try out the system yourself
after this presentation.**



Improvements

“ There's always a room for improvement. ”

MHM Rajpoot

Currently, both applications are running on the CPU without autograd from TensorFlow.

Secondly, the camera capturing the image is a laptop cam, which tends to capture a lot of noise.

With improvements in these two areas,

the results can potentially reach a perfect score of **5 out of 5**



Improvements :: No GPU and AutoGraph

```
WARNING:tensorflow:AutoGraph is not available in this env  
ome environments like the interactive Python shell. See h  
python/autograph/g3doc/reference/limitations.md#access-to  
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.  
Annotated (640, 480, 3)
```

GPUs leverage parallel processing can improve time required to verify from 30 to 10 sec.

Autograph in TensorFlow converts Python code into optimized graph operations, enhancing GPU utilization and overall computation speed.

Improvements :: Camera Quality



Both are 300x300

Professional cameras capture rich details, facilitating easy classification based on high-variance features.

Visit

[GitHub](#)

[YouTube](#)

Thanks
