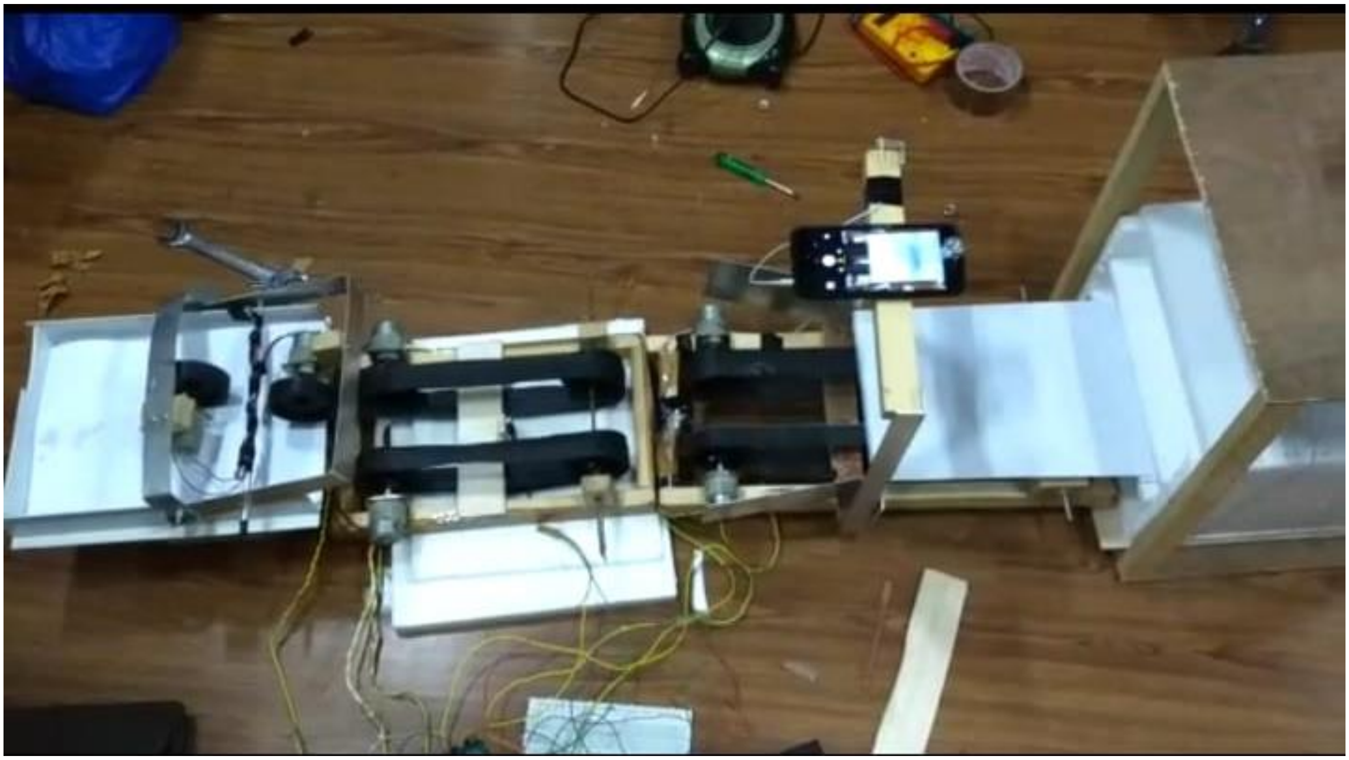
APPLICATION FORM READER
ITSP – 2018



Members-

1. Siddharth Chandak
2. Abhishek Tanpure
3. Jatin Lamba
4. Harsh Dolhare

Introduction-

Our project aims to automate the tedious work of manually reading various application forms. Our system will take in multiple forms, read them one at a time and digitalize the data (using Intelligent Character Recognition) and sort them into different labels. We require forms which have capital letters filled in separate blocks.

Parts of the Project-

This is a brief description of all the aspects of this project-

- Feeding Mechanism and the Conveyor Belt-
    1. Feeding Mechanism-

       Used to take in the top form one by one from a pile kept in an acrylic compartment which is elevated with wooden framework above conveyor belt. Has a seesaw like mechanism and is made using three motors, two to move paper onto the conveyor belt and one to change the seesaw state.
    2. Conveyor Belts-

       Two conveyor belts made using rubber sheets and rollers on a wooden framework. They consist of IR sensors attached to the wooden frame to detect the paper incoming to automate the process of its motion.
       The second conveyor belt also has a mobile stand for taking photos.

- Reading data from forms and storing it-
    1. Capturing Photo-

       We capture photos automatically using audio jack connected to a phone. We give signal using relay and an Arduino.
    2. Image Processing-

       Using Canny edge detection and Contour detection we extract all the boxes from the captured photos and read the letters from each box.
    3. Intelligent Character Recognition-

       We use convolutional neural network to read letter from each box. Image is resized into a 30*30 matrix which is then fed into the network as features. We are using separate classifiers for digit recognition and alphabet recognition.

- Sorting them according to different labels-

Each form is assigned a label after ICR is done according to the labels of each form which are described later. With the help of Processing IDE, we transfer the status to Arduino.
We made a mechanical model consisting of three slots made from polystyrene sheet which moves up and down according to the label of the incoming form using stepper motor with threads attached to slots.

Working of different parts of the Project-

1. Feeding Mechanism
   For this we used seesaw like mechanism with two 12 V DC motors and tires attached to them on the two sides of the seesaw such that only one of them is in contact with the paper below. One of the tires has more friction and takes one page forward and the other takes the pile backward which has moved a bit forward along with the top page due to static friction. The third motor is used to change the state of feeding mechanism (i.e. it decides which one of the above two motors will work at one time) using a see-saw mechanism. It uses a belt and pulley system to rotate the seesaw.
   The three motors are controlled using L298 motor driver IC which has 5 V operating voltage and can be signaled and controlled using Arduino. We used L298 IC instead of L293D IC as it has maximum 2A output current compared to just 0.6A of L293D IC.

2. Conveyor Belt and IR sensors
   There are two conveyor belts made using rubber sheets and rollers rotated using 12 V DC motors fixed on a wooden framework. The motors have to give a high torque to run the tight conveyor belt and hence required quite high input current. We used relays to switch the motors on and off instead of L298 motor driver IC as there was a significant output voltage drop across IC which reduced the current reaching the motors.
   It takes an individual paper from the feeding mechanism to a place where the photo is taken and then recollects them after this. During the sorting time it transfers the paper into various slots. It consists of three IR sensors in which the first one for detecting whether the paper has reached the first conveyer belt to order the feeding mechanism to change its state. The second IR sensor is used to stop the conveyor belt after the paper has reached the desired spot for taking the

photo , this is the only time when the conveyor has to stop , for rest of the time the conveyor belt is always turned on. The third IR sensor is used only in the sorting part. It gives the signal to the Arduino that the paper has gone into the corresponding slot for the next cycle to begin.

3. Capturing the Image

A photo of the form is taken using a mobile phone connected with a audio jack. The audio jack is shorted using a 5V relay by the signal from Arduino which then clicks a photo of the form placed beneath it just at the appropriate time (i.e. when the paper stops on the second conveyor belt). This set up works the same way as a selfie stick does when its button is pressed manually.

The image which is taken on the smartphone firstly gets uploaded on Google Photos automatically and then gets downloaded on the computer in a folder using a software called ‘Backup and Sync from Google’.

4. Extracting the boxes

For this we first open image of each form using Pillow package and then convert it to Grayscale using the OpenCV package. We have used OpenCV for all the Image processing functions. We apply Gaussian Blur on our image and detect the edges using Canny Edge Detection. Then we find contours in the image and then using the approxPolyDP function we get the 4 approximate corners of our page. Then using the four-point transform, we get a top view of the page. Then we binarize our image and store it. The above method is a simple version of the method used by apps like CamScanner etc.

After we obtain the scanned image, we again Gaussian Blur on it and then find contours in the image. We take only the contours which have area in a specified range. Then we sort those boxes according to their coordinates and identify what they correspond to (e.g. Name, Mobile Number etc.). Then we check if a box is empty by finding the number of pixels with intensity above some threshold.

5. Intelligent Character Recognition

Image of each character obtained above is fed into a neural network. Our network is like ImageNet network. We have two convolution layers followed by two Pooling layers. After that we have two fully connected layers. We have implemented the neural network in Keras library. We are also using dropout (0.2) to avoid overfitting on the training data. In convolutional layers we are using

3*3 filter with stride = 1. The training data that we used consisted of 2000 128*128 sized image of each letter and each digit.

6. Assigning status and transferring each form’s status to Arduino

If the number of boxes detected is less than the actual number of boxes, then we assign the status of manual rechecking to that form. If all the boxes of a certain field are empty, we assign the incomplete status to the form. If the all the boxes are identified and all fields are filled, we assign the complete status to the form. We store these labels as comma separated integers in a text file.

The label of each form needs to be sent from the text file to the Arduino via Arduino serial port for sorting. We are using Processing IDE to interact between the text file and Arduino serial port. It will send labels of the forms in that order which will be stored in Arduino as an array.

The link of similar data transfer has been given below.

7. Sorting forms according to their status

After taking photo and further processing of each form, the forms are collected and manually put back to the feeder part for the sorting process.

The set of slots moves up and down like a lift with the conveyor belt fixed that loads paper into the slots. A stepper motor is placed on the top of a wooden frame and will pull the set of polystyrene compartments up and down. We have used a 12 V bipolar stepper motor (JK42HS48-1684) with 4.4 kg-cm holding torque with DRV8825 motor driver IC which operates on 5 V and hence can be controlled using Arduino.

For each form, Arduino commands the stepper motor to rotate by specific degree according to the label of the topmost form such that the respective compartment gets aligned with the level of conveyor belt. The topmost form is then transferred on the conveyor belt using the feeding mechanism. The third IR sensor is fixed at the end of the second conveyor belt so that it makes sure that the form has reached its slot. It signals the Arduino when the current form reaches the corresponding compartment which begins the next cycle. Hence all forms are sorted into respective slots. The link for one such project has been given below.

Problems/Possible Limitations-

- Multiple Pages from Feeding Mechanism-
  Transferring one form at a time was one of the biggest challenges we faced. The first method we tried was the 3-roller mechanism used in the printer. But unfortunately, this method worked only if two pages went from the first roller. So, then we adopted the see-saw mechanism we are currently using. This also made sure that the tires remain in contact with the top paper despite the variable height of the pile of forms. But even now we can separate pages in only about 75% cases.
- Unable to extract all boxes-
  Extracting all the boxes is necessary for correctly reading the complete form. We used various image filtering techniques like blurring, dilation etc. Even if there was a small gap in a box, our algorithm sometimes skipped that box. Though we were able to achieve a detection rate of about 95%, we could not detect all 100% boxes.
- Problems in Character Recognition-
  We tried several standard OCR (like Tesseract) and some ICR models, but the CNN we trained ourselves is giving an accuracy greater than 75%. This is one major problem in our project we could not really solve.

References-

For sorting forms- https://naispace.wordpress.com/school-projects/mie444-mechatronics-principles/mie444-automated-paper-sorter-machine/
Dataset for English characters- https://www.nist.gov/srd/nist-special-database-19
Transferring form status to Arduino (from text file)-
https://arduinobasics.blogspot.com/2012/05/reading-from-text-file-and-sending-to.html
For scanning the page - https://www.pyimagesearch.com/2014/09/01/build-kick-ass-mobile-document-scanner-just-5-minutes/

Components required-

ITEM	**COST(Rs.)**
Acrylic sheet	240/-
Clamps	40/-
Battery	450/-
Arduino	450/-
Wheels	30/-
Threaded rod	80/-
IR Sensor	240/-
Arduino Cable	50/-
12 V DC motor	150/-
Jumper Cables	100/-
Polystyrene Sheet	-
Relay	-
Stepper Motor	721/-
DRV 8825(Stepper motor)	200/-