**Intelligent Character Recognition – Royal Sundaram Lifeline Insurance**

**Introduction:**

Intelligent character recognition (ICR) is a framework used to recognize and capture handwriting from scanned image forms. Manual data entry from hand-printed forms is very time consuming - more so in offices that have to deal with very high volumes of application forms (running into several thousands). Intelligent Character Recognition (ICR) System has the potential of improving efficiency in these offices using state-of-the-art technology.

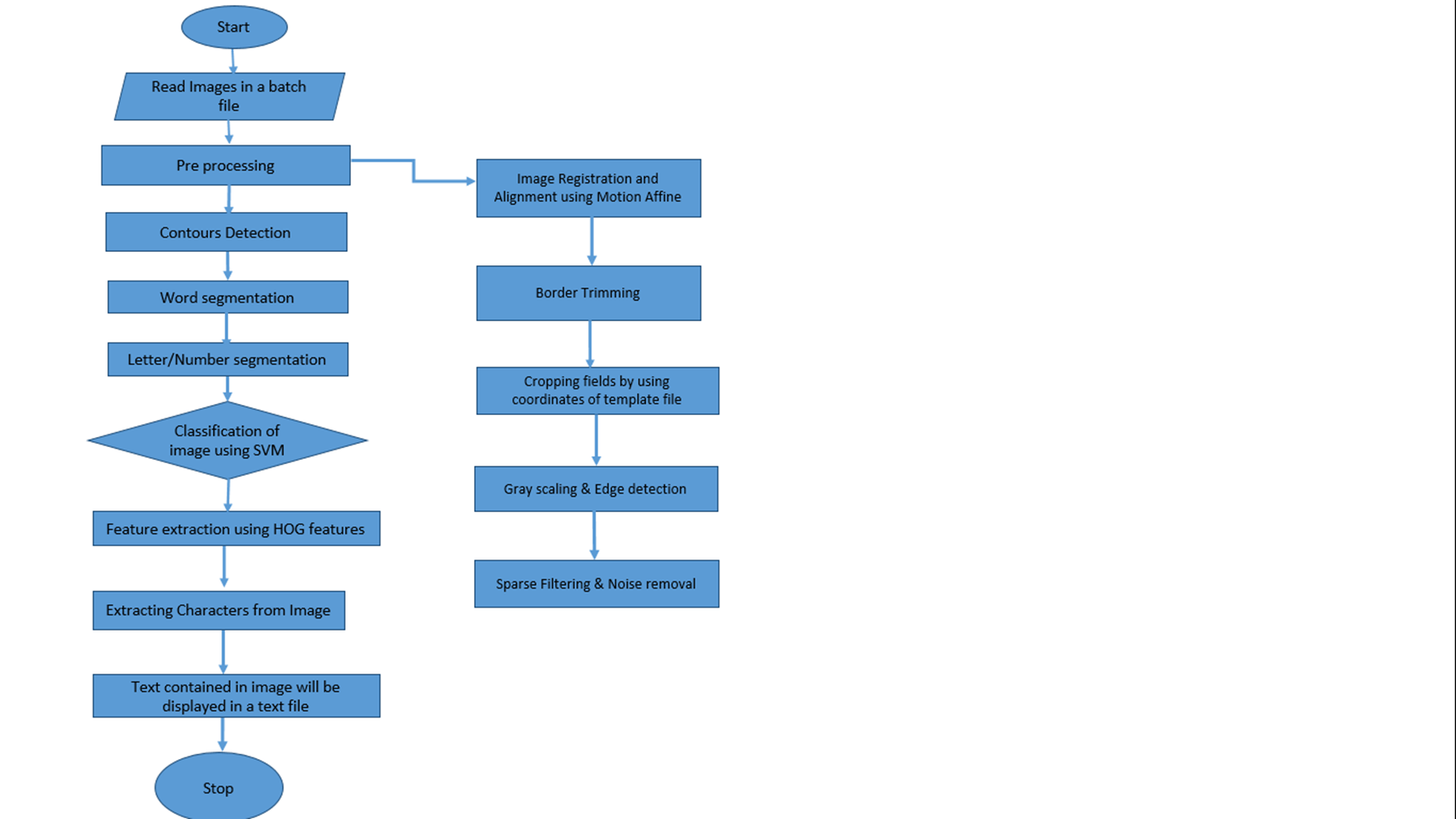
***The goal of the project is Intelligent Character Recognition (ICR) of handwritten data in the scanned insurance forms provided by insurance company and to automate the process for digitization of forms.***

**Current Business Challenges for Digitization:**

* Ambiguous Data - Form Scanning Quality, Data Entry Not Consistent
* Data Entry Errors - Quality Issues
* Time Consuming And Process Complexity
* Employee Attrition
* Rising Data Volumes- IT Bandwidth
* Rising Cost Of Manpower- Diminishing Margins

**Workflow of KAL ICR Solution:**

**Flow Chart of KAL ICR Solution:**

****

**Note:**

* Currently we are using CNN for digit/character recognition
* For CNN, no need of using external feature extraction methods before modelling/ prediction
* Work flow of the solution is same irrespective of model (CNN/SVM) we were used

**Broad Steps Involved in KAL ICR Automated Framework:**

* Data Preprocessing/Preparation
  1. Reading the handwritten scanned forms (PDF/TIF) and converting them into JPEG/JPG images
  2. Selecting the Template form
  3. Identifying the different categories(numeric/character/alpha numeric/checkbox) in the form and creating the coordinate dictionary for each field in the Image
  4. Border detection using Homography
  5. Image Alignment using Affine transformation
  6. Image difference using Background Subtraction
  7. Cropping the different categories in the each form
  8. Median smoothing of each cropped field
  9. Finding contours and drawing bounding rectangles for each cropped field
* Modeling for Image Recognition (Independent Activity – No dependency on the other steps)

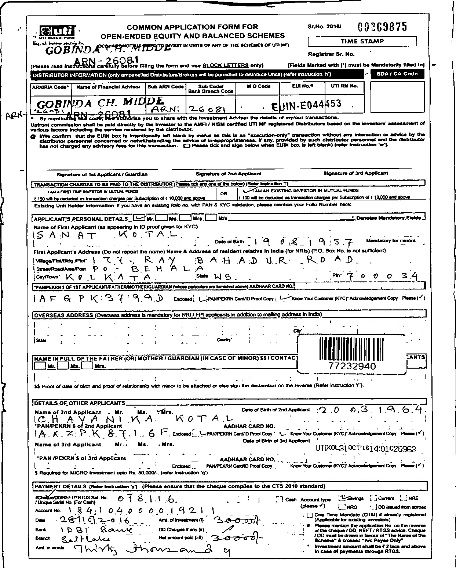
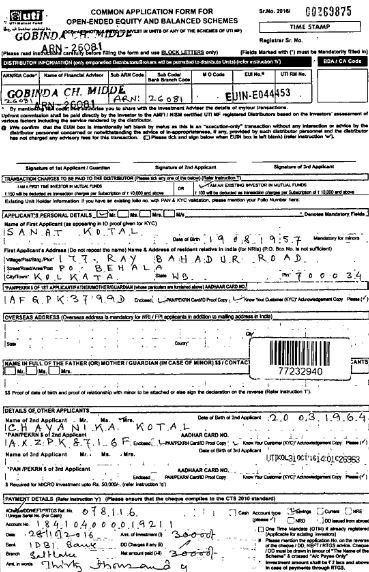
1. Collect the MNIST/NIST images from the web sources
2. Building SVM digit/character/alphanumeric recognition model by extracting the HOG features from the Images
3. Building CNN digit/character recognition model
4. Building the SVM model for predicting the 2-length checkbox fields

* Output Extraction
  1. In case of SVM, Need to extract HOG features from each bounding rectangle and recognizing the digits/characters/alpha numeric
  2. In case of CNN, Need to only recognize the digits/characters/alpha numeric
  3. Predict the 2-length checkbox fields using Checkbox predictive modeling and >2-length checkbox fields using high-pixel intensity logic
  4. Combine the predictions for each field
* Future Scope

**Description of Steps involved:**

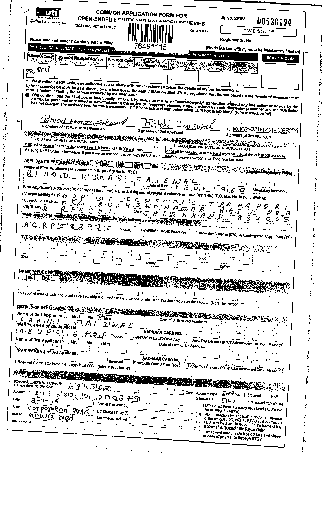
* **Data Preprocessing/Preparation:**

1. **Reading the Images:** Read the scanned insurance forms from the directory using ***tiftojpeg/ pdftojpeg*** function and convert them to JPEG/JPG images
2. **Selecting the Template form:** Select the template form which is having properties like high dpi, proper alignment, border visibility and noise free etc.,
3. **Identifying the categories and creating the coordinate dictionary for fields:** identify the different categories and corresponding fields in the form and create the coordinate dictionary for each field using pixlr/paint.
4. **Border detection using Homography:** Detect the border using image Homography technique (***border\_detect*** function in the code).border\_detect function internally depends on ***whitespace*** and ***non*** ***whitespace*** functions

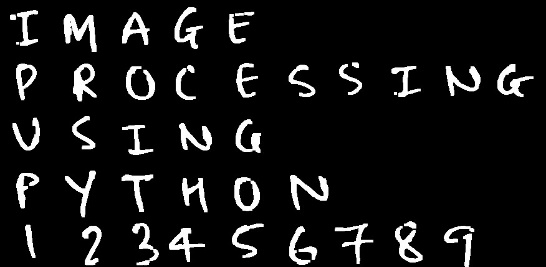
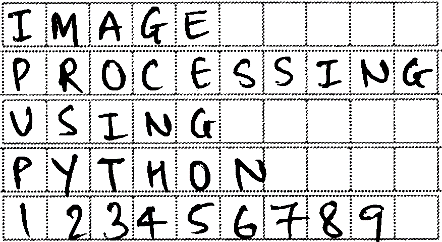
**Original image Image after border detection**

1. **Image Alignment using Affine transformation:** Align the each image using warpAffine transformation (***align*** function in the code). align function internally depends on ***register\_affine*** function

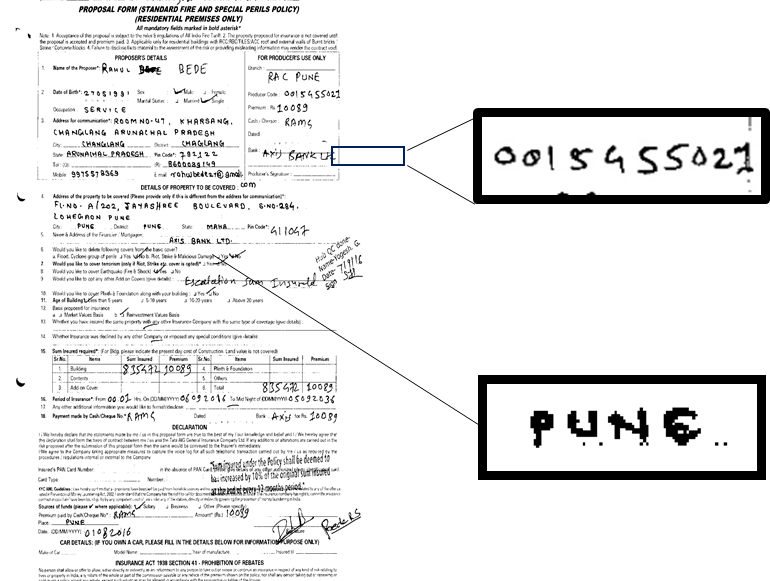
**Before alignment After alignment**

1. **Image difference using Background Subtraction:** Perform the background subtraction by subtracting the template image from the filled image (***image\_difference*** function in the code). This operation will be helpful in removing grids/lines from the image



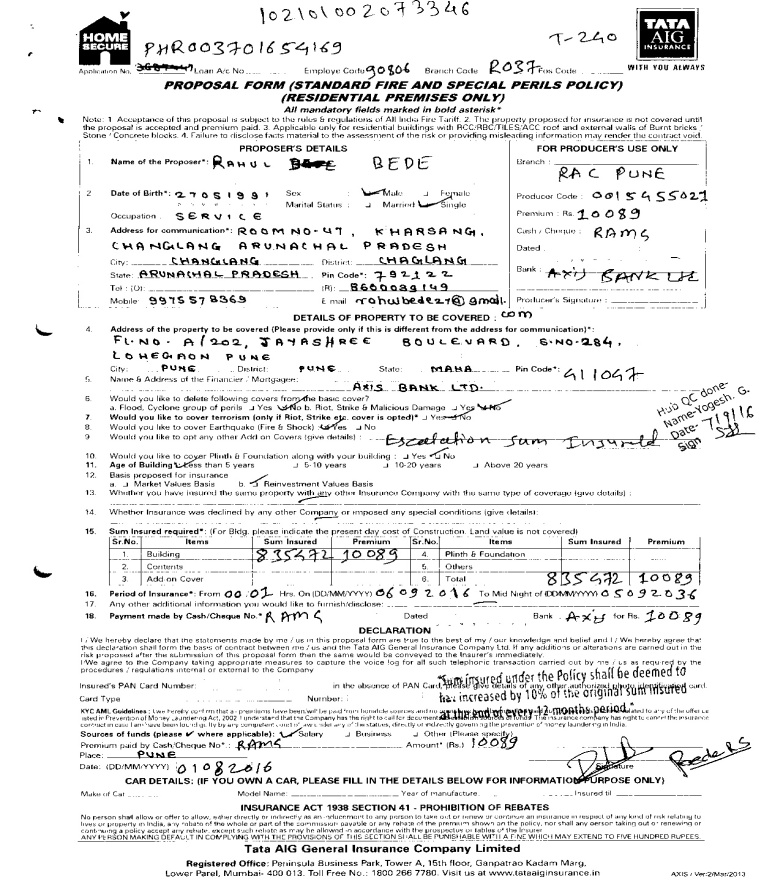
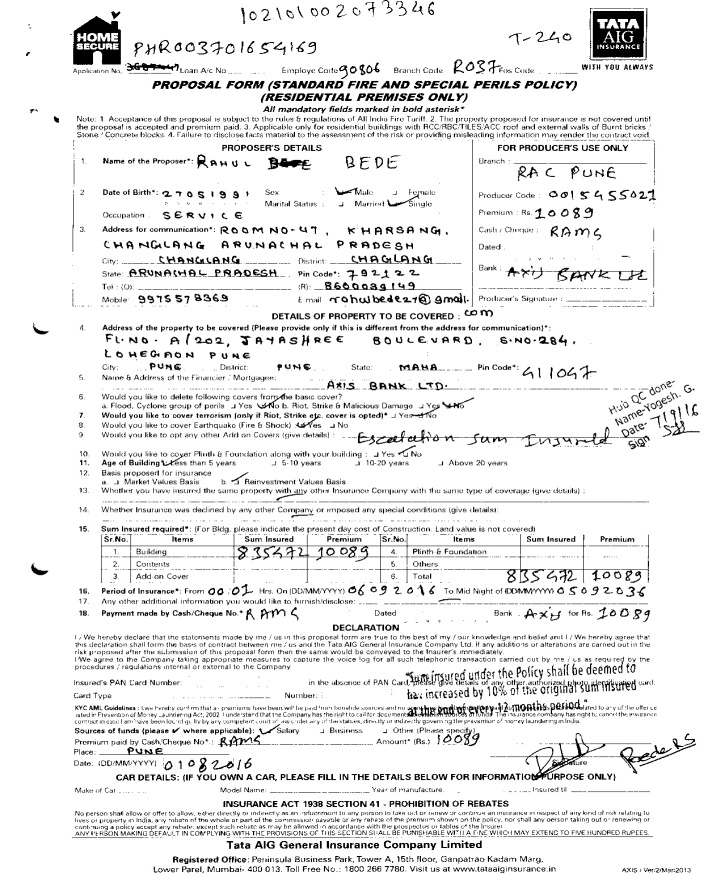
**Original image Background subtracted image (inverted)**

1. **Cropping the different categories in the each form:** Crop the different fields in the image using coordinates dictionary (***cropping*** function in the code)

****

**Note:**  We will use background subtracted images for cropping the individual fields

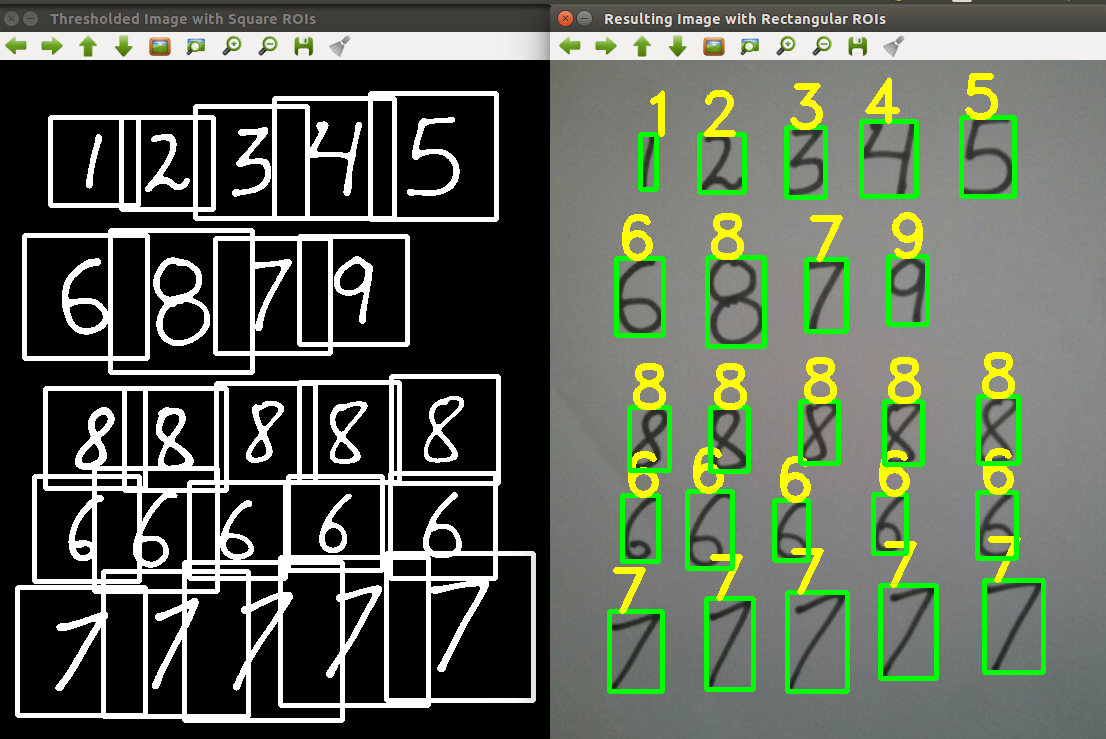
1. **Median smoothing of each cropped field:** Apply the median filtering (cv2.median function in ***output\_cray*** function) on each field for removing salt and pepper noise from the image.

**Original Image Median Image**

**Note:** We have to perform this operation on each cropped field images

1. **Finding contours and bounding rectangles:** Finding the contours from the cropped field and identify the bounding rectangles (cv2.findContours and cv2.boundingRect functions in ***output\_cray*** function) from the contours.

****

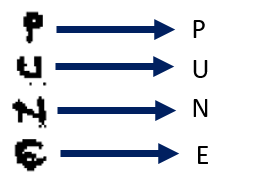
* **Modeling for Image Recognition:**

1. **Collect the MNIST/NIST images from the web sources:**  Collect/download the MNIST/NIST images from the web links (digits/lower case/upper case alphabets)
2. **Build the SVM model:** Build the separate SVM models (using HOG features/independent variables extracted from the image) for digits, characters and alpha numeric separately
3. **Build the CNN model:** Build the separate CNN models for digits, characters and alpha numeric separately
4. **Build the SVM model for 2-length Checkbox fields:** Building the 2-length checkbox predictive model by using pixel intensities as features

* **Output Extraction:**

1. **Predictions on Numeric/Character/Alpha numeric fields using SVM model:**  Extract the HOG features from the bounding box and predict the each bounding box image using SVM model (refer ***output\_cray*** function in the code)
2. **Predictions on Numeric/Character/Alpha numeric fields using CNN model:** Predict the each bounding box image using CNN model (refer ***output\_cray*** function in the code)
3. **Predictions on Checkbox fields using SVM model:** Predict the 2-length checkbox fields using Checkbox predictive modeling and >2-length checkbox fields using high-pixel intensity logic (refer ***output\_cray*** function in the code)
4. **Combine the Predictions:** Combine the prediction of each bounding boxs for each field

**Example:**

****

**Bounding boxes and Image Recognition**

**Cropped Field**

PUNE

**Extracted Output**

**Output Extraction**

* **Future Scope:** 
  1. Automatic grid removal for all kind of images irrespective of image quality
  2. Image Alignment when there is no border in the Image
  3. Code Optimization
  4. Reducing the Form extraction time

**Path for Codes/Forms/Pickle Files/Documents of Royal Sundaram Lifeline Insurance:**

[**\\192.168.209.12\karvy analytics\KT-Sai**](file:///\\192.168.209.12\karvy%20analytics\KT-Sai)