Breaking Squirrel Mail Captcha

Group:- MLG4
CS771A (Machine Learning)

Bhangale Pratik Anil : 14173
Ashish Kumar Singh: 14142
Gaurav Kumar : 14240
Bhargav Ganguly: 14177
Shubham Jain: 14676

Problem Statement

Design a ML algorithm that will correctly recognize a string of characters in a given CSE Squirrel Mail / IITK Webmail captcha image.

CAPTCHA stands for ’Completely Automated Public Turing test to tell Computers and Humans Apart’.

CSE online Squirrel Mail / IITK Webmail requests a text based captcha.

Existing Approach

- Ideally, the following steps are followed for captcha breaking
  - Localize characters in the image
  - Obtain characters using Segmentation
  - Character recognition
- Various methods are used for last step of character recognition ranging from naive to advanced methods
  - Multi Class Classification of characters using SVM
  - CNN's to learn features of characters and perform recognition
- Certain advanced methods (Goodfellow *et al* *) proposes to merge all the preprocessing steps by inputting entire images to deep neural nets

Our Approach

- Segmentation is used to extract characters out of image and finally CNN is used to learn and recognize characters.
- Process of segmentation varies for different captcha types depending on level of noise and clutter in image. It can be easy for CSE Captchas and quite tricky for noisy captchas such as IITK Webmail.
- Particular details about each captcha will be discussed subsequently in greater depth

Datasets

Datasets for Webmail1 and Webmail were grabbed from web page and manually labelled.

1. Webmail - 5000 labelled images
2. Webmail1 - 1200 labelled images
3. CSE - 30000 labelled images (thanks to group MLG35)

Note: There might have been some errors due to manual labelling

Different Captcha Type Examples

Webmail1





Webmail




CSE Squirrel Mail





Webmail1 Captcha

Key Observations:

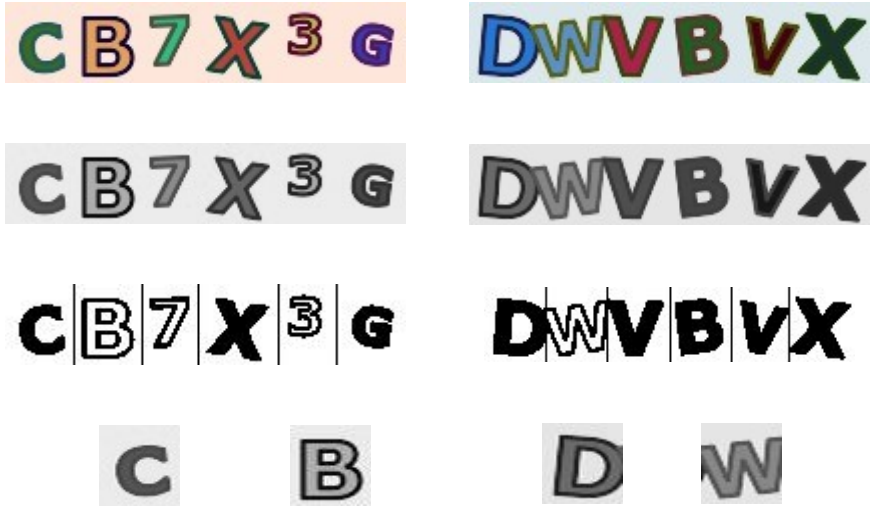
- No noise in background
- Always 6 characters are present
- Background is always lighter than character border
- Minor overlapping has been observed
- Characters are not warped (distorted)
- Image dimensions (40 x 200) pixels

Webmail1 Captcha Segmentation

Step1: Gray scaling and Thresholding
Convert the image into binary format and remove the background by thresholding

Step2: Segmenting image into 6 parts
Find vertical lines separating all individual characters

Step3: Resizing every image segment
Resize every segment to 40x40 pixels



CSE Captcha

Key Observations:

- There is noise in the background. (:P Really)
- **All noisy pixels have the same pixel value (colour).**
- Number of characters is always 5.
- Characters are not warped (distorted).
- No rotation in the characters.
- All character pixels have the same value (colour).
- Background is always white.

CSE Captcha Segmentation

Step1: Noise removal
All noise pixels were modified to white pixels directly.

Step2: Grayscale and thresholding

Step3: Segmenting into 5 parts

Step4: Resizing every segment
Resize every image segment to 40x40 pixels

Webmail Captcha

Observations:

- No generalized pattern of background noise
- Number of characters is either 3 or 4.
- Characters are not warped (distorted).
- No rotation in the characters.
- Characters are visually almost equally spaced

Binary Classification for length Webmail Captcha

- Entire Captcha is input to a CNN network

Architecture:

- The network has 3 convolutional layers where each convolutional layer is followed by a max pooling operation and ReLU activation function.
- Max Pool operation has a stride window of size 2 X 2.
- 3 fully connected layers are deployed after these convolution operations where the final layer is output layer.
- Output layer has 2 nodes corresponding to two classes . One class corresponds to captcha of length 3 and other to length 4 captcha
- Network returns number of characters present in captcha.
- Loss function used is Cross Entropy Loss.

Segmentation of Webmail Captcha

1. Work with the noisy background
2. Try to remove background noise, two broad way to extract letters :
    1. Try to get the boundary of letter precisely - **Boundary Segmentation**
    2. Separate the color of the letter inside the boundary - **Dominant color segmentation**

Webmail Captcha Segmentation I

Step1: Binary Classify the image to get number of characters

Step2: Equally divide the image into desired number of segments

Step3: Resizing every segment
Resize every image segment to 80x60 pixels

Webmail Captcha Segmentation II

Boundary segmentation

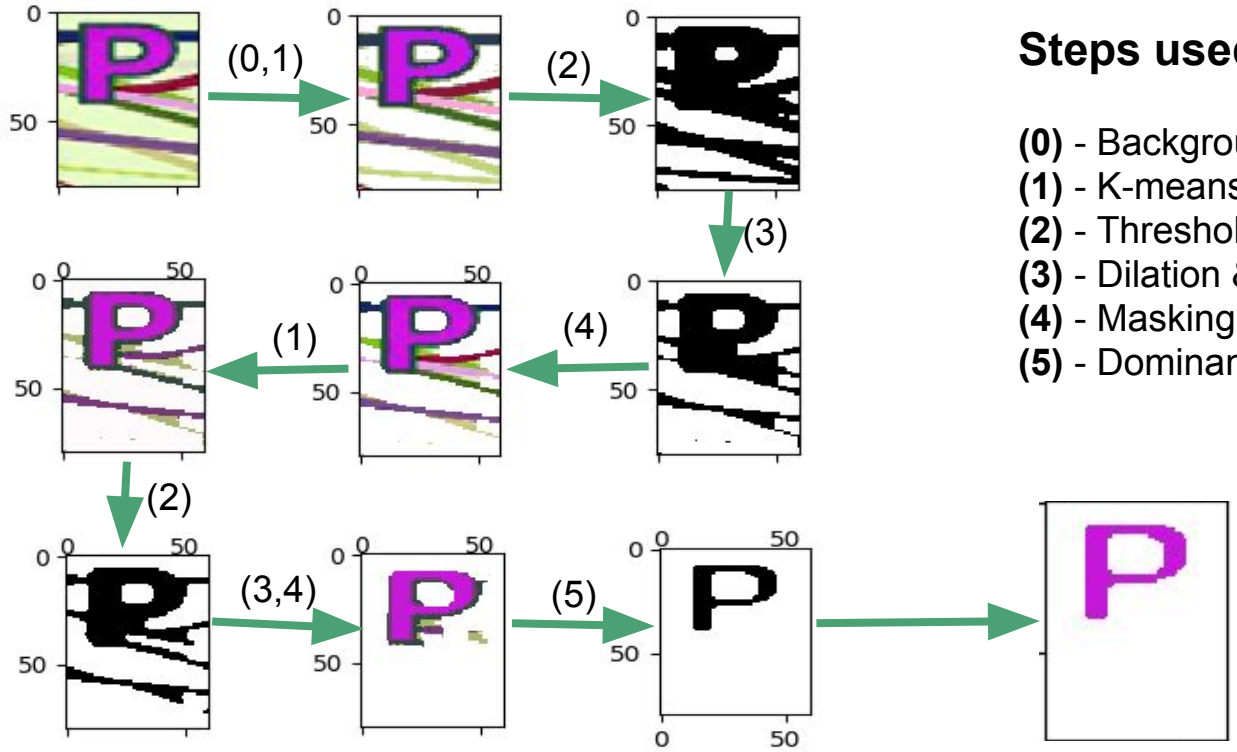
- Set threshold on the gray scale image of the captcha
- Remove connected components with less than a 25 points
- Filtering with morphological filters
- Apply this mask on coloured image

Webmail Captcha Segmentation III

Dominant color segmentation

- Background color is filtered out from coloured image
- Thresholding, dilation and connected component segmentation is used on gray scale image to get the mask
- Mask is applied on coloured image
- K-means clustering is used on the masked image for the color quantization
- Dilation and erosion sequentially applied to further smooth the image
- Finally dominant color is extracted out to get the letter

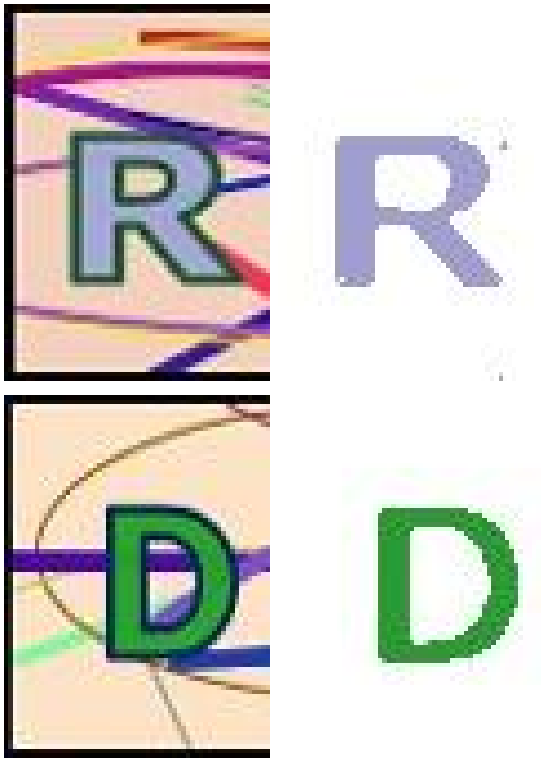
Segmentation III in action :-)



Steps used :

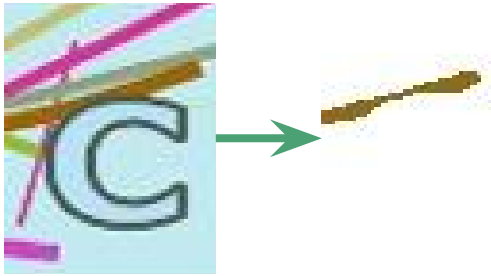
(0) - Background Removal
(1) - K-means Clustering
(2) - Thresholding
(3) - Dilation & Erosion
(4) - Masking
(5) - Dominant Color Extraction

Some sample outputs



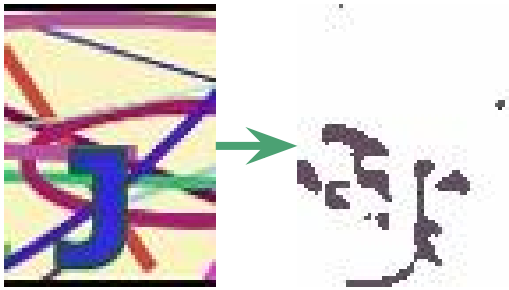
Some failed cases

- When background and image letter is of same color



- When Noise of same color having more area than actual character

  (Purple color has more area than that of Blue)



Overall around **90%** images correctly captured (By observation)

Convolutional Neural nets (CNN)

- The segmented character images are sent to a CNN for recognition

Architecture:

- The network has 2 convolutional layers where each convolutional layer is followed by a max pooling operation and ReLU activation function.
- Max Pool operation has a stride window of size 2 X 2.
- 3 fully connected layers are deployed after these convolution operations where the final layer is output layer.
- Output layer has nodes equal to number of classes and is used for prediction.
- Loss function used is Cross Entropy Loss

Experimental Results

1. **CSE Webmail:**

- Training dataset size:30,000
- Testing dataset size:10,000
- Train Accuracy:100%
- Test Accuracy:100%

Note: This accuracy is w.r.t individual characters not entire CAPTCHA

Experimental Results

2. **Webmail1:**

- Training dataset size:1,000
- Testing dataset size:200
- Train Accuracy:100%
- Test Accuracy:97%

Note: This accuracy is w.r.t individual characters not entire CAPTCHA

Experimental Results

Webmail (Binary Classification for length):

- Training dataset size:4,000
- Testing dataset size:1,000
- Train Accuracy:99%
- Test Accuracy:99%

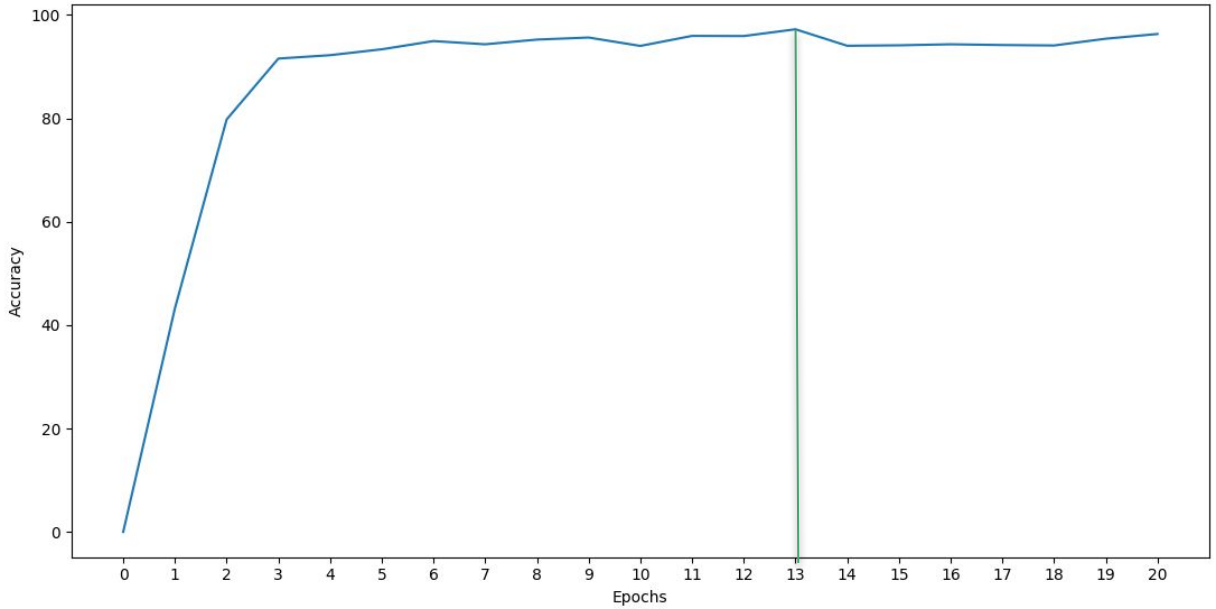
Experimental Results

3. **Webmail (Direct Method):**

- Training dataset size:4,000
- Testing dataset size:1,000
- Individual Characters Train Accuracy:99%
- Individual Characters Test Accuracy:96%
- Whole Captcha Test Accuracy:80.7%

Parameters Tuning

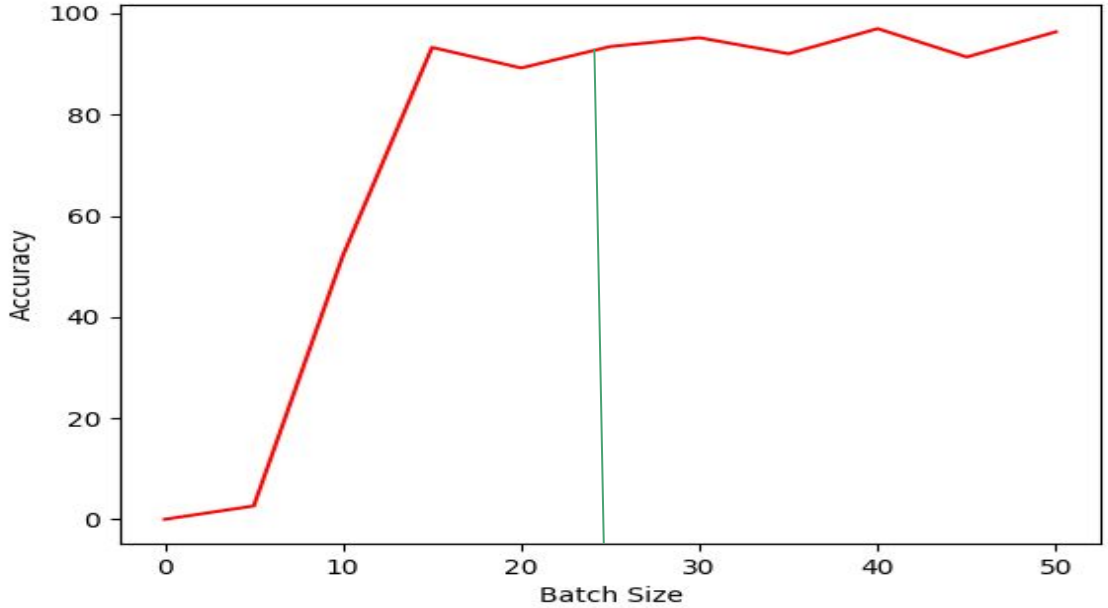
Epochs (Number of times training was repeated on same dataset)



Datasets
Training: 3000 images
Validation : 1000 images

Parameters Tuning

Accuracy vs Batch size



Datasets
Training: 3000 images
Validation : 1000 images

Experimental Results

4. **Webmail (Boundary Segmentation approach):**

- Training dataset size:4,000
- Testing dataset size:1,000
- Train Accuracy:97%
- Test Accuracy:94%

Note: This accuracy is w.r.t individual characters not entire CAPTCHA

Experimental Results

5. **Webmail (Dominant Colour Segmentation approach):**

- Training dataset size:4,000
- Testing dataset size:1,000
- Train Accuracy:96%
- Test Accuracy:91%

Note: This accuracy is w.r.t individual characters not entire CAPTCHA

Future work

- Finding a workaround for labelling the dataset.
- Finding a way for dealing with Captchas which may have one or more characters colored same as the background color.
- Finding a way to deal with large noisy lines that cut through the character and have same color as the character.