Vietnam's Motor Plate Recognition

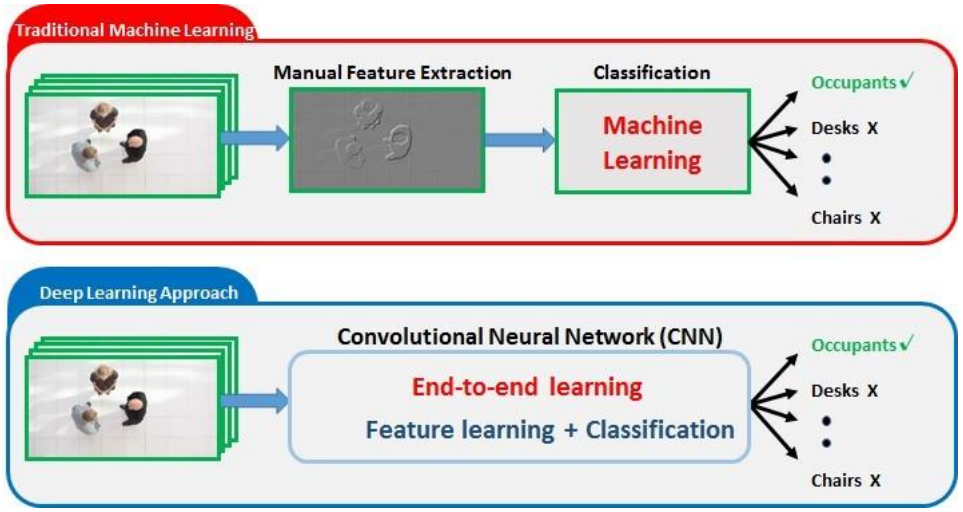
Presenter: Nguyen Dinh Tung – R&D Lab

Human vs Machine

- Machine doesn't create a new way to solve problem.
- Human need to define the route and machine will help to get there quickier

Trade-off in real life

- Trade-off in real-life
  - Child vs Adult
  - Computer vs Super Machine



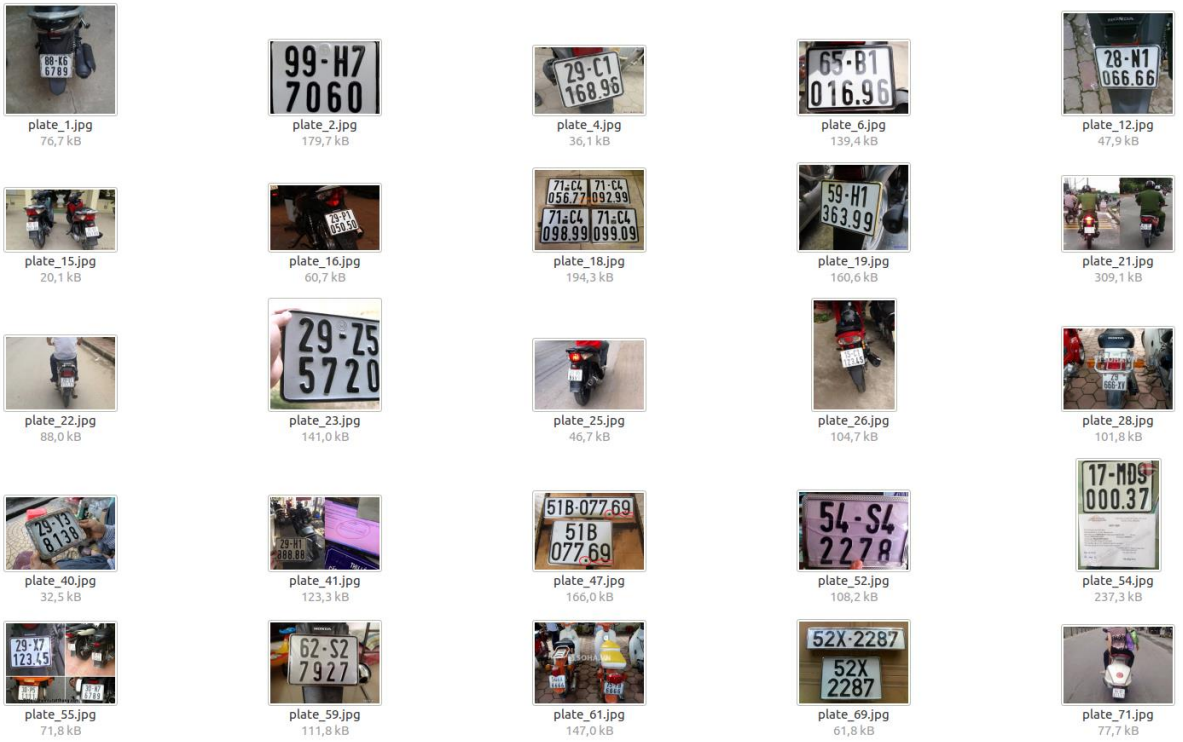
- Methodology
  - Environment => Number
  - Environment => Plate => Number
  - Environment => Char => Number
  - Environment => Plate => Char => Number

Outlines

1. Data Collection
2. Plate Recognition
3. Char Separation
4. Char Recognition
5. Limitations and Improvements

Data Collection – Plate

- Google Image Download
  - pip install google_images_download
  - googleimagesdownload --keywords "bien so xe may" --limit 2000
- Scrapy || BeautifulSoup
  - More applicable for shopping/ organized websites



Data Collection - Chars

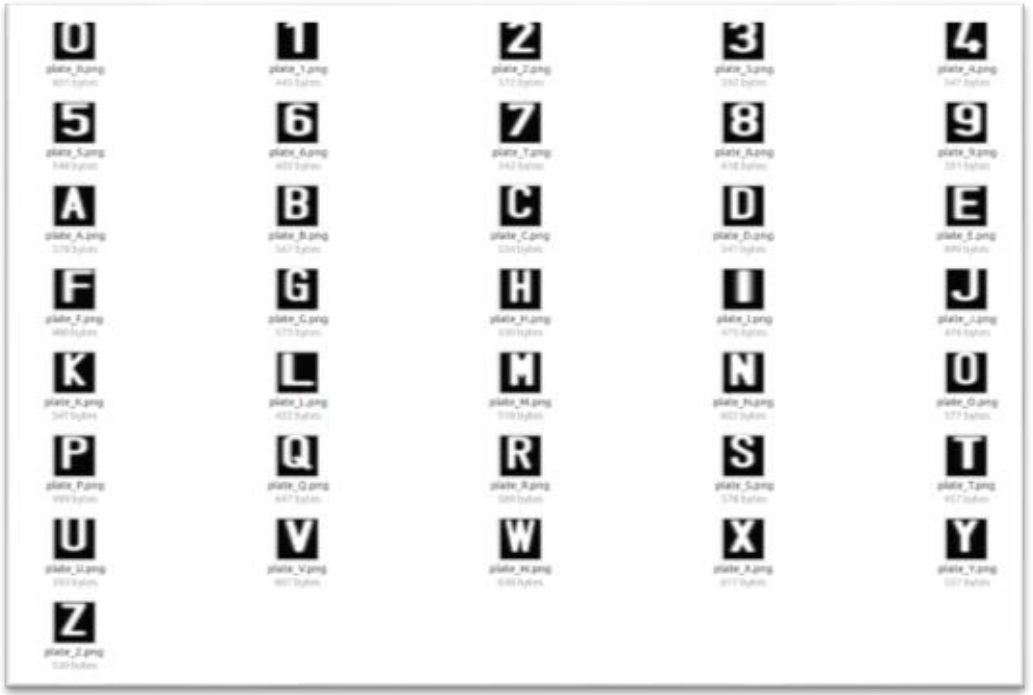


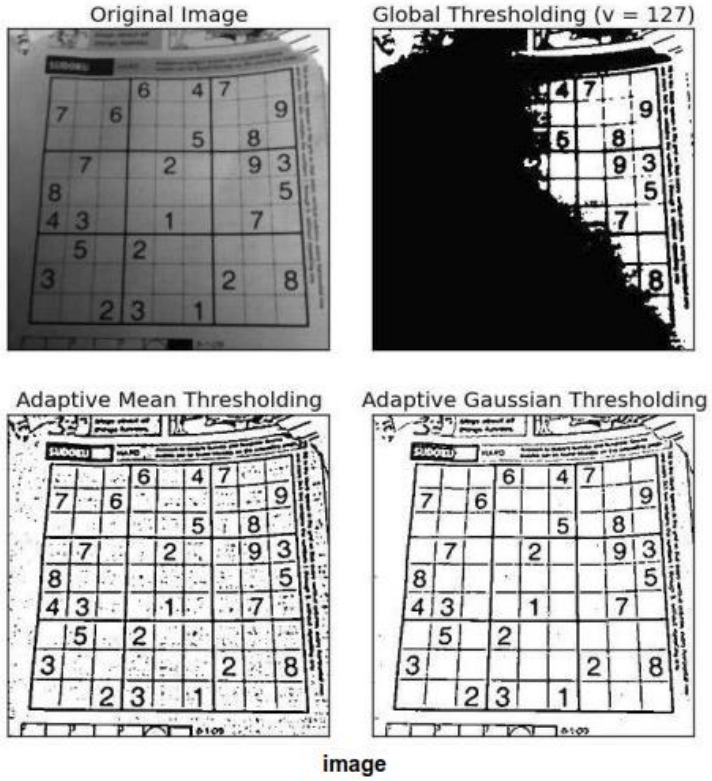
Plate Recognition

Original

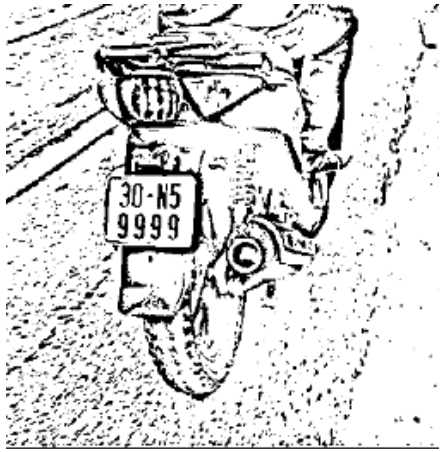
cv2.cvtColor

cv2.adaptiveThreshold

Adaptive threshold



Binary Threshold



AdatptiveThreshold

Plate Recognition

cv2.findContours

List of Contours

List of list of Contours

Plate

Char Segmentation

Plate

Black & White

Threshold

cv2.GaussianBlur

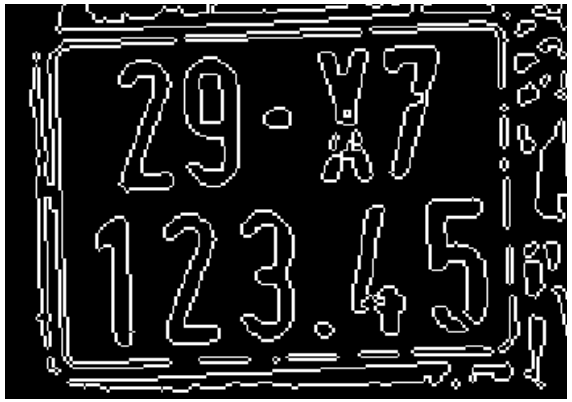
Contours

Suitable Contours

Segmentation

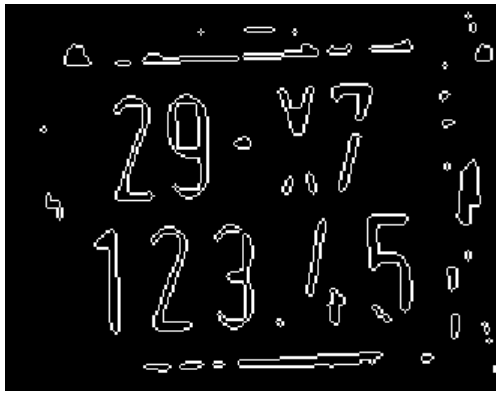
Blur Params

Adaptive Threshold



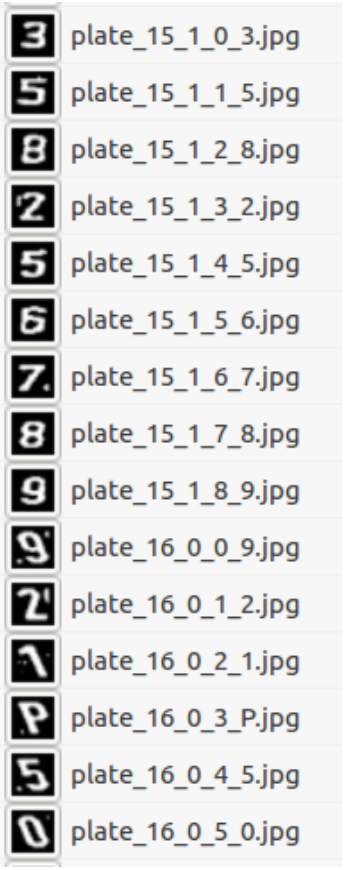
Blur(3, 3)

Blur(5, 5)



Blur(9, 9)

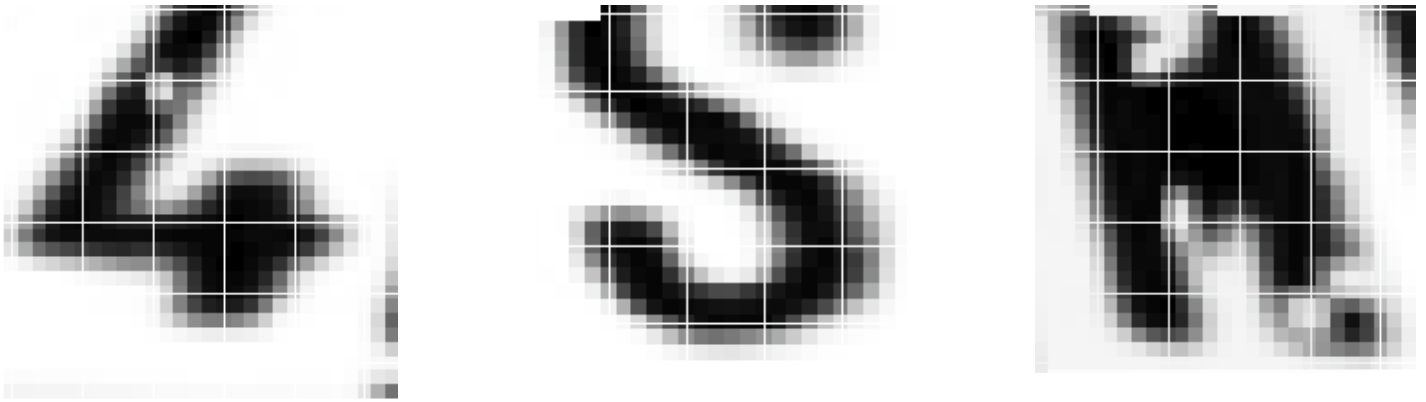
Char Recognition - Train Data



- Extract Character
- Use a simple char recognition
- Re-label 2000 images
- Unable to recognize 300 images

Char Recognition – Aug Data

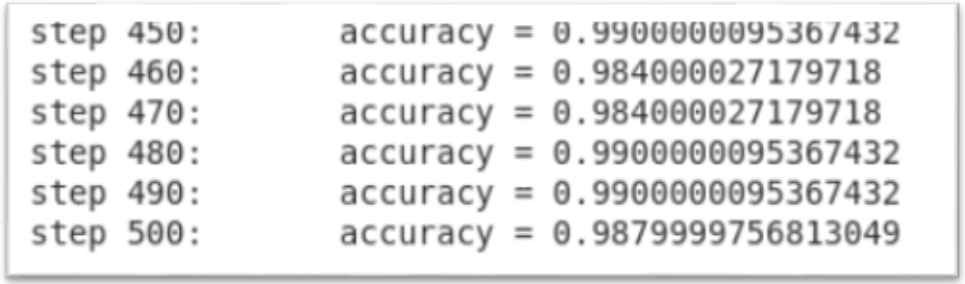
- Source: Real Data + Alphabet Generating (sum = 1734)
- Outputs: Augumented Data => 173.400 images



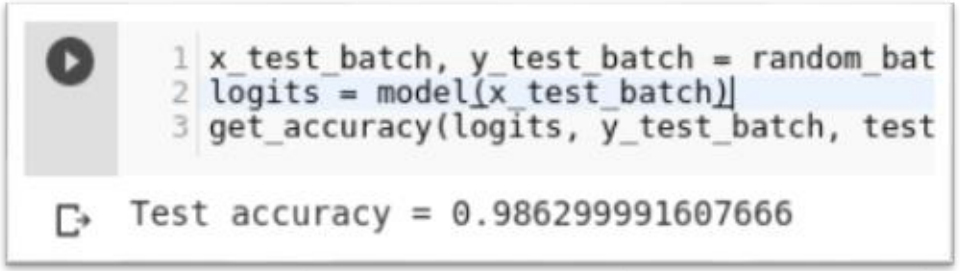
Char Recognition

- Google Colab
  - CSV file
  - Limitation in RAM and GPU: 12GB, K80
- Techniques:
  - Tensorflow Eager Mode
  - CNN model:
    - Conv2D(32, 5, activation = tf.nn.relu, padding = "SAME")
    - MaxPooling2D(2, 2)
    - Conv2D(64, 5, activation = tf.nn.relu, padding = "SAME")
    - MaxPooling2D(2, 2)
    - Flatten()
    - Dense(256, activation = tf.nn.relu)
    - Dropout(0.75)
    - Dense(num_classes, activation = None)

	0	1	2	3	4	5	6	7	8	9	...	775	776	777	778	779	780	781	782	783	784
0	C	0	0	0	0	0	1	1	0	0	...	0	0	0	0	0	0	0	0	0	0
1	5	0	1	31	133	217	177	141	215	254	...	0	0	0	0	0	0	0	0	0	0
2	T	67	2	1	2	8	24	47	77	101	...	0	0	0	0	0	0	0	0	0	25
3	1	0	0	0	0	0	1	13	14	14	...	1	0	0	0	0	0	0	0	0	0
4	D	0	18	95	195	247	253	253	241	185	...	0	0	0	0	0	0	0	0	0	0



```
step 450:       accuracy = 0.9900000095367432
step 460:       accuracy = 0.984000027179718
step 470:       accuracy = 0.984000027179718
step 480:       accuracy = 0.9900000095367432
step 490:       accuracy = 0.9900000095367432
step 500:       accuracy = 0.9879999756813049
```

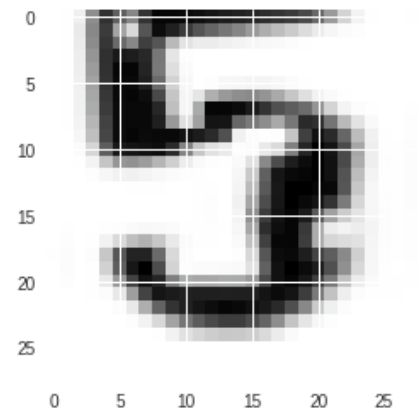


```
1 x_test_batch, y_test_batch = random_bat
2 logits = model(x_test_batch)
3 get_accuracy(logits, y_test_batch, test
```

```
Test accuracy = 0.986299991607666
```

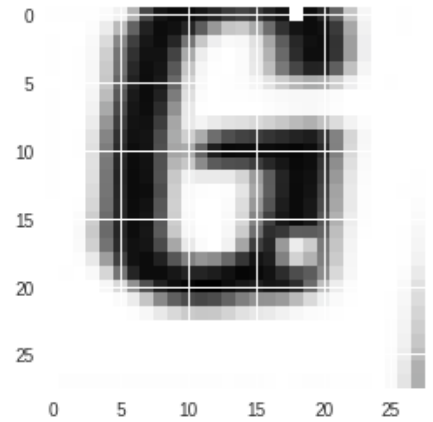
Char Recognition

```
# Check image source
i = 1
im = np.asarray(data.iloc[i, 1:], dtype=np.float32).reshape(28,28)
plt.imshow(im)
plt.show()

logits = model(im)
y = tf.argmax(tf.nn.softmax(logits), axis = 1)
print("\nThe Character is:", le.inverse_transform(y.numpy()))
```



```
The Character is: ['5']
```

```
# Check image source
i = 23
im = np.asarray(data.iloc[i, 1:], dtype=np.float32).reshape(28,28)
plt.imshow(im)
plt.show()

logits = model(im)
y = tf.argmax(tf.nn.softmax(logits), axis = 1)
print("\nThe Character is:", le.inverse_transform(y.numpy()))
```



```
The Character is: ['G']
```

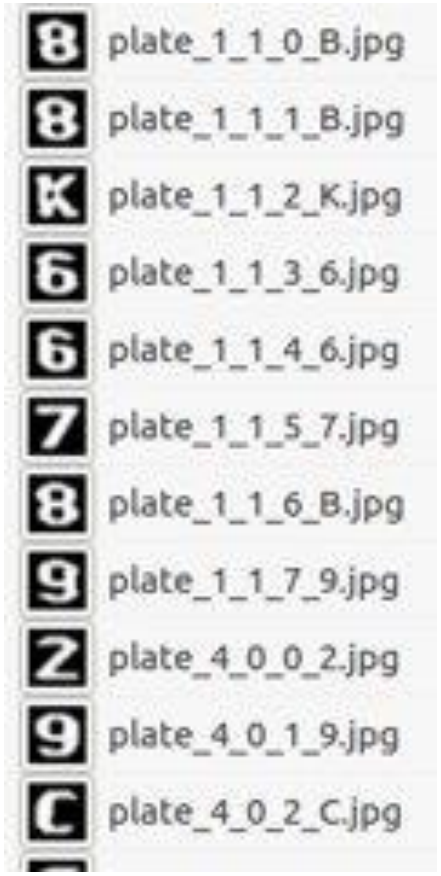
Results

Results


30-N5
9999
30559999


65-D1
111.11
65-B1
111.11
65-P6
1111
65D111111
658111111
65P61111

Difficulties & Limitations



- Difficulties
  - Hand coded: hyper params, blur input chars
  - Lack of memory, GPU
  - Mislabel
  - Wrong label: 1-7, H-N, 8-B, 0-D
- Step Results
  - Plate Recognition: 90%
  - Char Segmentation: 85%
  - Char Recognition: 98%

Improvements

- Plate Recognition Steps:
  - Colour/ Shape - Problems?
  - Multiple methods?
- Char Segmentation – Error Analysis
- Environment => Plate => Number
  - Relation among Characters: 30M6-2804, 52H2-6666, 55P5-3432
- Environment => Char => Number
  - Specific plate material

Q&A

- Q means Questions
- A means Answers

- (Q => A) && (!Q => !A)
- !A => !Q?
- A => Q?

Useful Link

- My Github: https://goo.gl/vKkttH
- This Slide: https://goo.gl/tCU8yY

Thank you for your listening

Command

- Gen Chars
  - python GenChars.py --font True
- Separate Chars
  - python Main.py gen ./test/train_data/ ./fonts/real_chars/
- Augment Chars
  - python GenChars.py --gen_char 10
- Detect Chars
  - python Main.py detect ./test/train_data/plate_2.jpg --steps True --save True