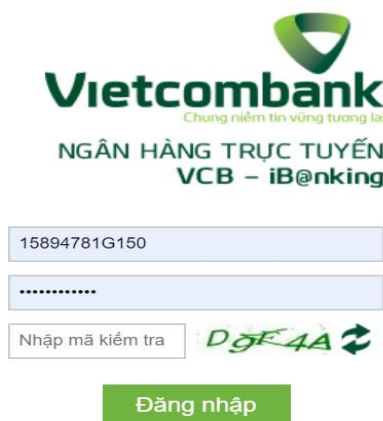


# How a simple Tensorflow based program defeat CAPTCHA from almost Vietnamese banks

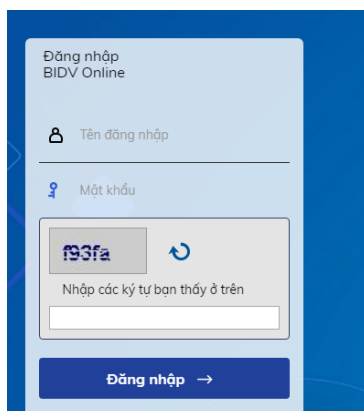
Nguyễn Hồng Điệp  
Product Manager, ketoan.app

# Captcha là gì?



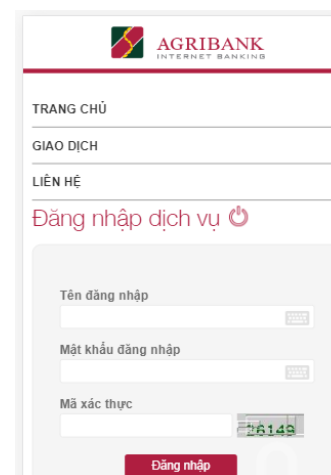
The image shows the login page of Vietcombank. At the top is the Vietcombank logo with the tagline "Chung niềm tin vững tương lai" and "NGÂN HÀNG TRỰC TUYẾN VCB - iB@nking". Below the logo are two input fields: the first contains the number "15894781G150" and the second contains a series of dots. Below these fields is a label "Nhập mã kiểm tra" (Enter verification code) and a CAPTCHA image showing the text "D9F4A" with a refresh icon. At the bottom is a green button labeled "Đăng nhập" (Login).

**VIETCOMBANK**



The image shows the login page for BIDV Online. It has a blue header with the text "Đăng nhập BIDV Online". Below the header are two input fields: the first is labeled "Tên đăng nhập" (Username) and the second is labeled "Mật khẩu" (Password). The password field contains the text "f93fa" and a refresh icon. Below the password field is a label "Nhập các ký tự bạn thấy ở trên" (Enter the characters you see above) and a CAPTCHA image showing the text "f93fa". At the bottom is a blue button labeled "Đăng nhập" (Login) with a right arrow.

**BIDV**

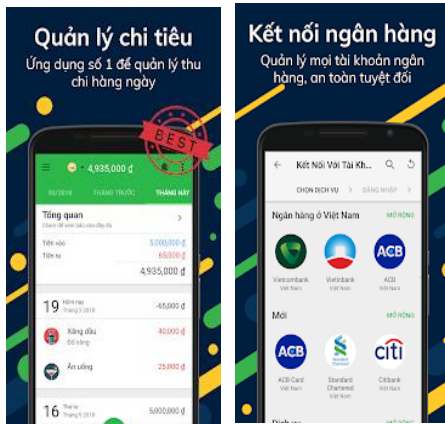


The image shows the login page for AGRIBANK Internet Banking. It has a white header with the AGRIBANK logo and the text "INTERNET BANKING". Below the header are three links: "TRANG CHỦ" (Home), "GIAO DỊCH" (Transactions), and "LIÊN HỆ" (Contact). Below these links is a red button labeled "Đăng nhập dịch vụ" (Login service) with a refresh icon. Below the button are three input fields: the first is labeled "Tên đăng nhập" (Username), the second is labeled "Mật khẩu đăng nhập" (Login password), and the third is labeled "Mã xác thực" (Verification code). The verification code field contains the text "26149" and a CAPTCHA image showing the text "26149". At the bottom is a red button labeled "Đăng nhập" (Login).

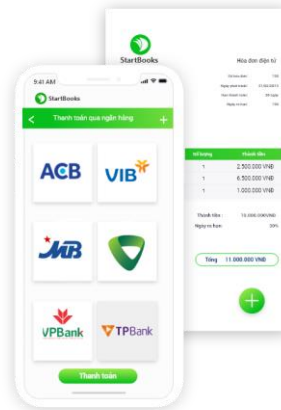
**AGRIBANK**

# Vượt captcha để làm gì?

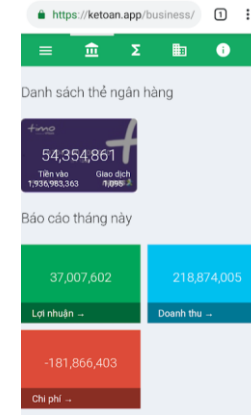
## MONEY LOVER



## MISA STARTBOOK



## KETOAN.APP



Accounting Automation Software

Credit Rating

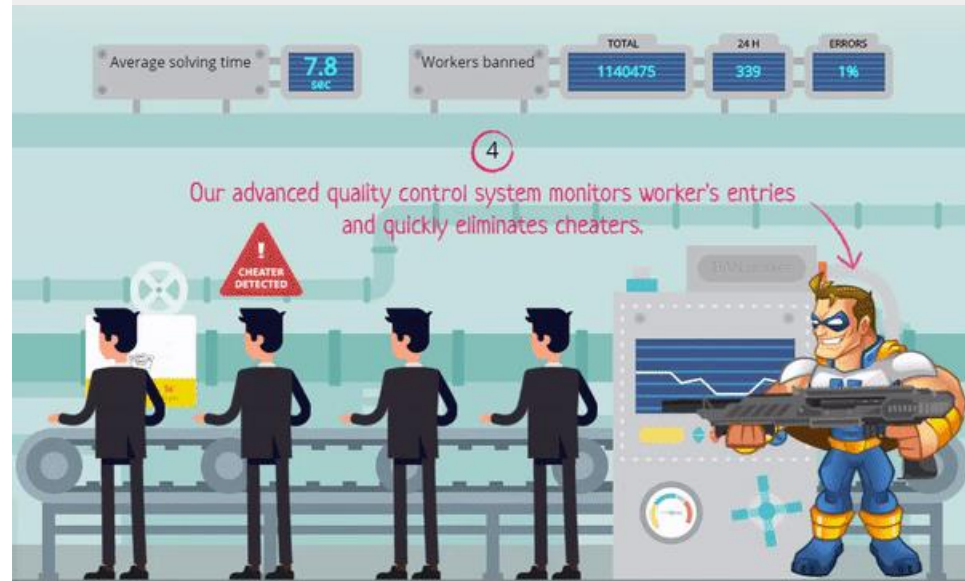
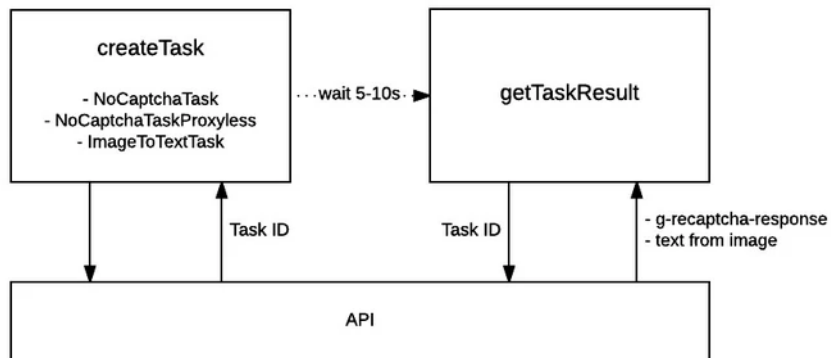
Bank Aggregator

Bots

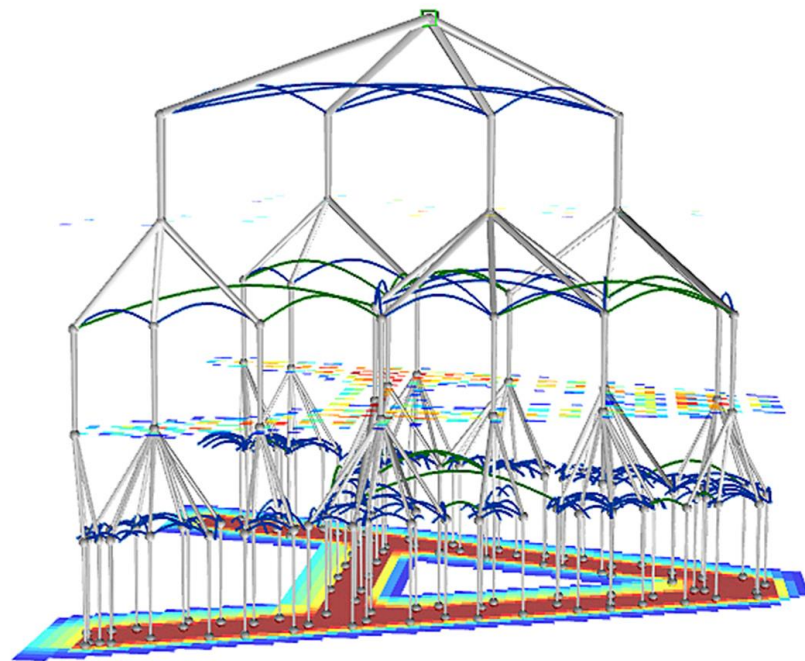
Captcha Bypass

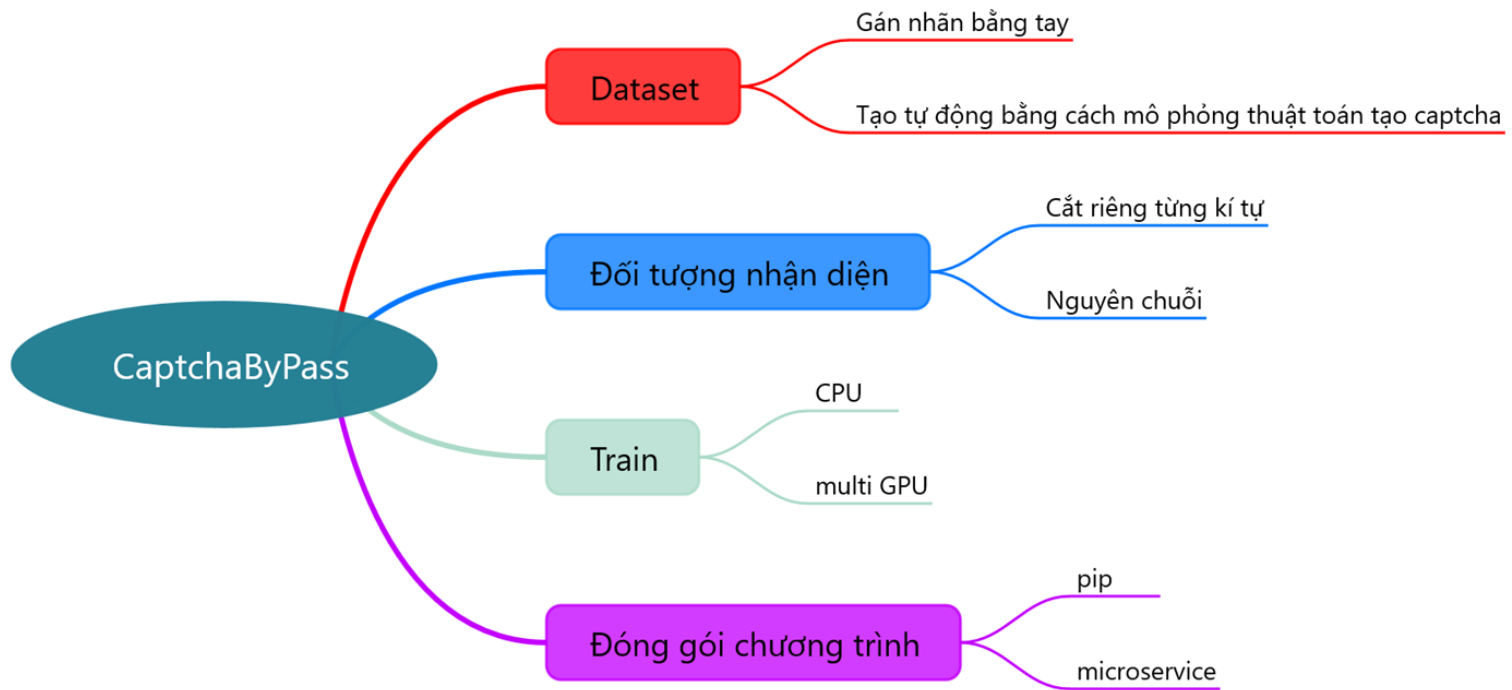
# Có một nghề gõ captcha

## Anticaptcha API



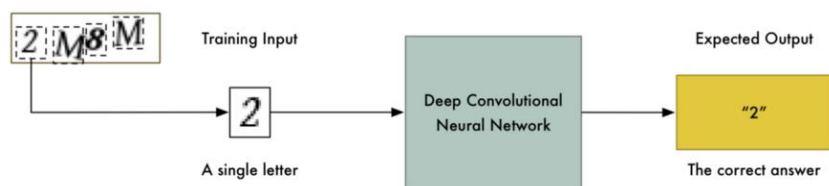
# Vượt Captcha với AI



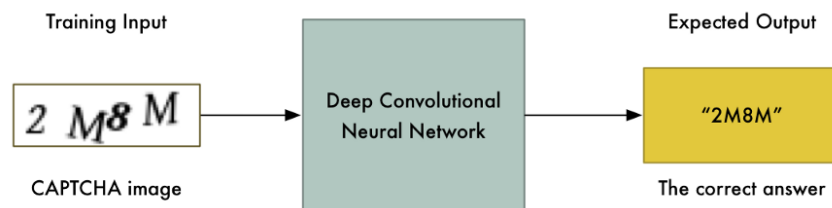


# Segmentation

## Theo từng kí tự



## Theo nguyên cụm



# Don't reinvent the wheel

Ngó qua github, khá nhiều opensource đã có sẵn.

The screenshot shows a GitHub search results page for the query 'captcha tensorflow'. The page header includes the GitHub logo, navigation links like 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing', and a search bar containing 'captcha tensorflow'. Below the header, there's a sidebar with filters for 'Repositories' (123), 'Code', 'Commits' (19), 'Issues' (91), 'Marketplace', 'Topics', 'Wikis' (12), and 'Users'. A 'Languages' section lists Python (91), Jupyter Notebook (10), HTML (4), JavaScript (3), Go (2), Java (2), and R (1). The main content area displays '123 repository results' sorted by 'Best match'. Three repositories are visible: 1. 'Tensorflow' by TensorFlow, described as an open source software library for numerical computation, with 462 stars. 2. 'JacksonYang/captcha-tensorflow', a Python repository for solving image captchas using TensorFlow and CNN, with 462 stars. 3. 'zhengwh/captcha-tensorflow', an HTML repository for hacking captchas using CNN with TensorFlow, with 208 stars. 4. 'nickliqian/cnn\_captcha', a Python repository for using CNN to recognize captchas with TensorFlow, with 612 stars.









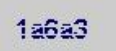





Dễ dàng tìm được

## 3 dự án captcha bypass (Tensorflow)

chất lượng tốt

trên  **GitHub** .

# vnbankpass

	
	No captcha
	
	
	
	
	No captcha

```
BANKS = {
  "vietcombank" : {
    "IMAGE_HEIGHT" : 25,
    "IMAGE_WIDTH" : 87,
    "CHAR_SETS" : '0123456789ABCDEF',
    "CLASSES_NUM" : 16,
    "CHARS_NUM" : 5,
  },
  "agribank" : {
    "IMAGE_HEIGHT" : 35,
    "IMAGE_WIDTH" : 75,
    "CHAR_SETS" : '0123456789abcdef',
    "CLASSES_NUM" : 16,
    "CHARS_NUM" : 5,
  },
  "bidv" : {
    "IMAGE_HEIGHT" : 35,
    "IMAGE_WIDTH" : 75,
    "CHAR_SETS" : '0123456789abcdef',
    "CLASSES_NUM" : 16,
    "CHARS_NUM" : 5,
  },
  "acb" : {
    "IMAGE_HEIGHT" : 27,
    "IMAGE_WIDTH" : 100,
    "CHAR_SETS" : '0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ',
    "CLASSES_NUM" : 36,
    "CHARS_NUM" : 5,
  },
  "vpbank" : {
    "IMAGE_HEIGHT" : 30,
    "IMAGE_WIDTH" : 150,
    "CHAR_SETS" : '0123456789',
  },
}
```

# vnbankpass

Để train cho một loại captcha mới

5,000

dataset  
(10USD)

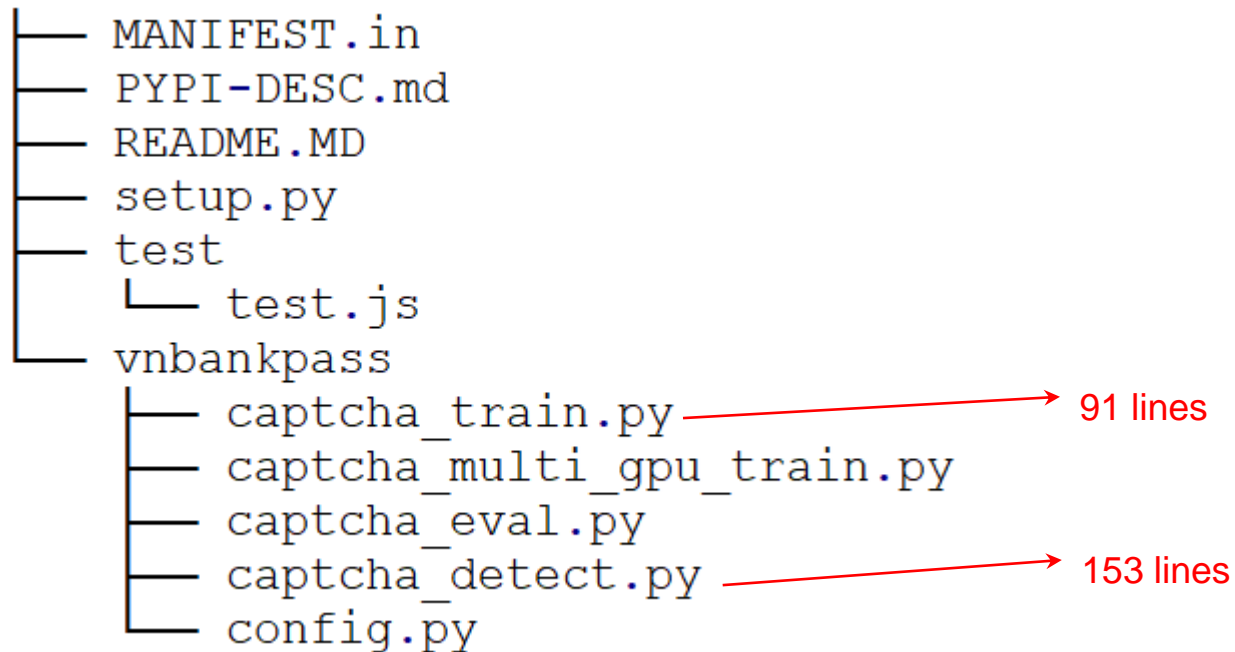
2 giờ

training time  
(Single CPU Desktop)

90%

độ chính xác

# Code structure



# CNN Model

```
images = tf.reshape(images, [-1, IMAGE_HEIGHT, IMAGE_WIDTH, 1])

with tf.variable_scope('conv1') as scope:
    kernel = _weight_variable('weights', shape=[3,3,1,64])
    biases = _bias_variable('biases', [64])
    pre_activation = tf.nn.bias_add(_conv2d(images, kernel), biases)
    conv1 = tf.nn.relu(pre_activation, name=scope.name)

pool1 = _max_pool_2x2(conv1, name='pool1')

with tf.variable_scope('conv2') as scope:
    kernel = _weight_variable('weights', shape=[3,3,64,64])
    biases = _bias_variable('biases', [64])
    pre_activation = tf.nn.bias_add(_conv2d(pool1, kernel), biases)
    conv2 = tf.nn.relu(pre_activation, name=scope.name)

pool2 = _max_pool_2x2(conv2, name='pool2')

with tf.variable_scope('conv3') as scope:
    kernel = _weight_variable('weights', shape=[3,3,64,64])
    biases = _bias_variable('biases', [64])
    pre_activation = tf.nn.bias_add(_conv2d(pool2, kernel), biases)
    conv3 = tf.nn.relu(pre_activation, name=scope.name)

pool3 = _max_pool_2x2(conv3, name='pool3')

with tf.variable_scope('conv4') as scope:
    kernel = _weight_variable('weights', shape=[3,3,64,64])
    biases = _bias_variable('biases', [64])
    pre_activation = tf.nn.bias_add(_conv2d(pool3, kernel), biases)
    conv4 = tf.nn.relu(pre_activation, name=scope.name)

pool4 = _max_pool_2x2(conv4, name='pool4')

with tf.variable_scope('local1') as scope:
    batch_size = images.get_shape()[0].value
    reshape = tf.reshape(pool4, [batch_size, -1])
    dim = reshape.get_shape()[1].value
    weights = _weight_variable('weights', shape=[dim, 1024])
    biases = _bias_variable('biases', [1024])
```

```
diepnh@PathenBot: ~/Projects/captcha/captcha_recognize
Search Terminal Help
08:02:49.182211 Step 178: true/total: 85/100 precision @ 1 = 0.850
08:02:49.877238 Step 179: true/total: 85/100 precision @ 1 = 0.850
08:02:50.541728 Step 180: true/total: 86/100 precision @ 1 = 0.860
08:02:51.212323 Step 181: true/total: 88/100 precision @ 1 = 0.880
08:02:51.897518 Step 182: true/total: 85/100 precision @ 1 = 0.850
08:02:52.853635 Step 183: true/total: 92/100 precision @ 1 = 0.920
08:02:53.938250 Step 184: true/total: 94/100 precision @ 1 = 0.940
08:02:55.153496 Step 185: true/total: 88/100 precision @ 1 = 0.880
08:02:56.211212 Step 186: true/total: 88/100 precision @ 1 = 0.880
08:02:57.215565 Step 187: true/total: 80/100 precision @ 1 = 0.800
08:02:58.362319 Step 188: true/total: 80/100 precision @ 1 = 0.800
08:02:59.475233 Step 189: true/total: 91/100 precision @ 1 = 0.910
08:03:00.848879 Step 190: true/total: 81/100 precision @ 1 = 0.810
08:03:02.137492 Step 191: true/total: 91/100 precision @ 1 = 0.910
08:03:03.403192 Step 192: true/total: 85/100 precision @ 1 = 0.850
08:03:04.513720 Step 193: true/total: 89/100 precision @ 1 = 0.890
08:03:05.841481 Step 194: true/total: 87/100 precision @ 1 = 0.870
08:03:06.744200 Step 195: true/total: 91/100 precision @ 1 = 0.910
08:03:07.516835 Step 196: true/total: 82/100 precision @ 1 = 0.820
08:03:08.573271 Step 197: true/total: 83/100 precision @ 1 = 0.830
08:03:09.985259 Step 198: true/total: 86/100 precision @ 1 = 0.860
08:03:11.148907 Step 199: true/total: 86/100 precision @ 1 = 0.860
08:03:11.148963 true/total: 17303/20000 precision @ 1 = 0.865
bot:~/Projects/captcha/captcha_recognize$
```

# Áp dụng AI vào product của bạn.

