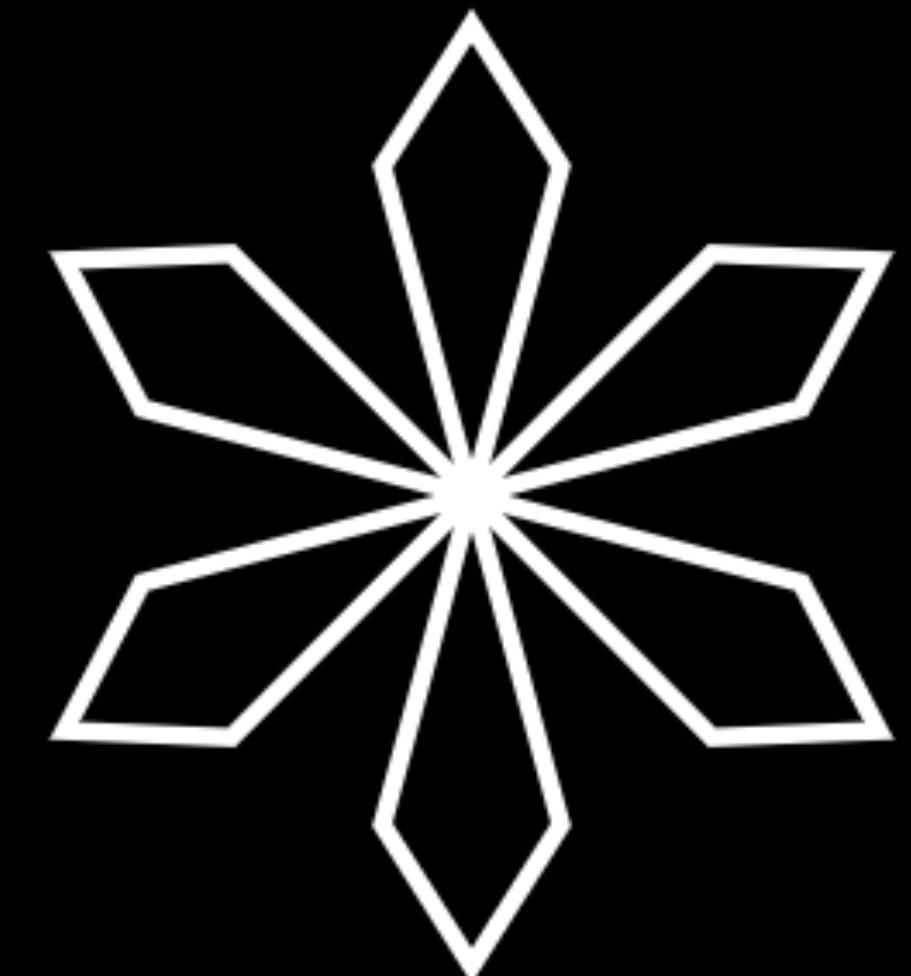


@NataliePis

# Captcha Challenge

#1

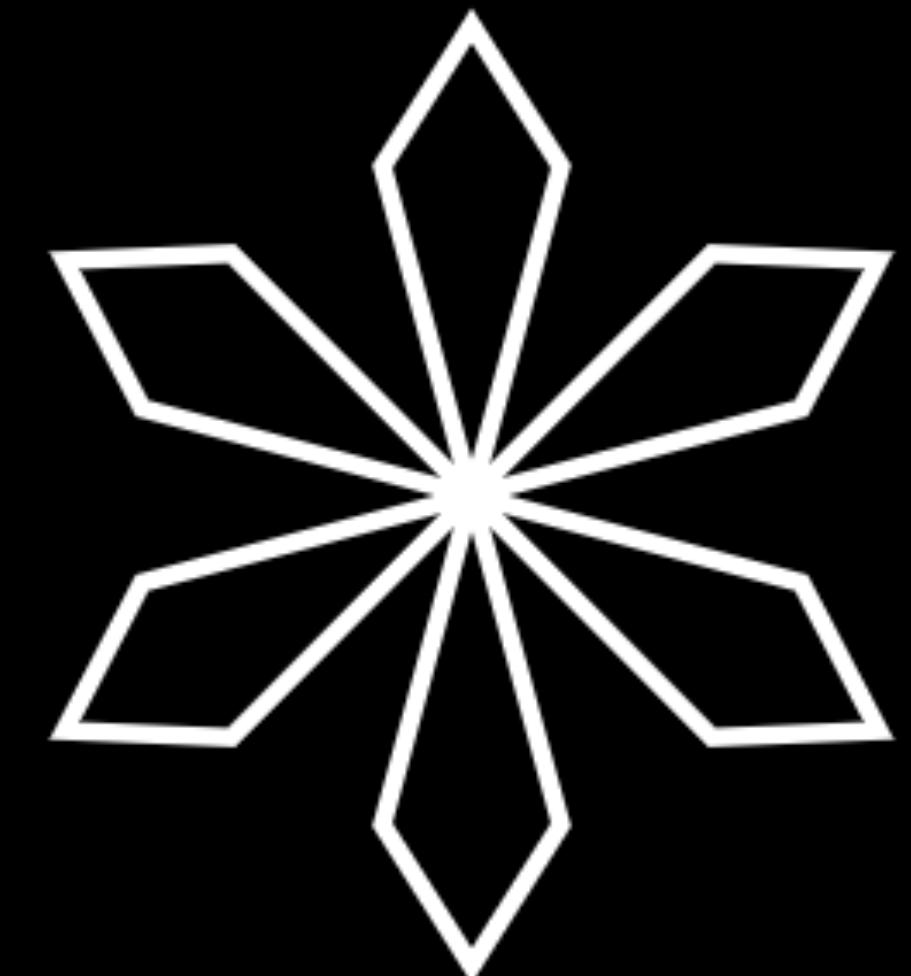




BLACKLIGHT

[blacklight.ai](https://blacklight.ai)

@blacklightai

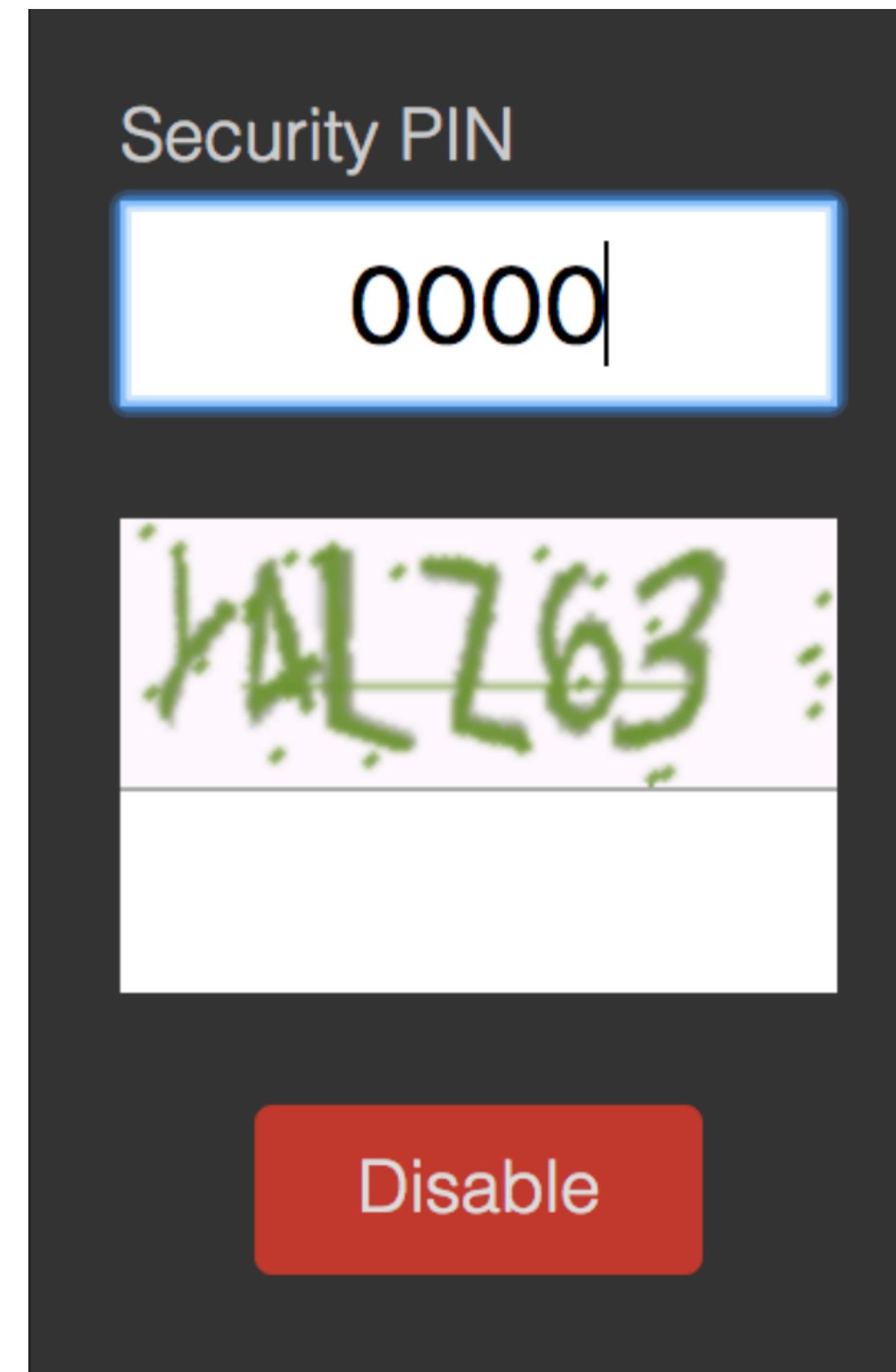


BLACKLIGHT

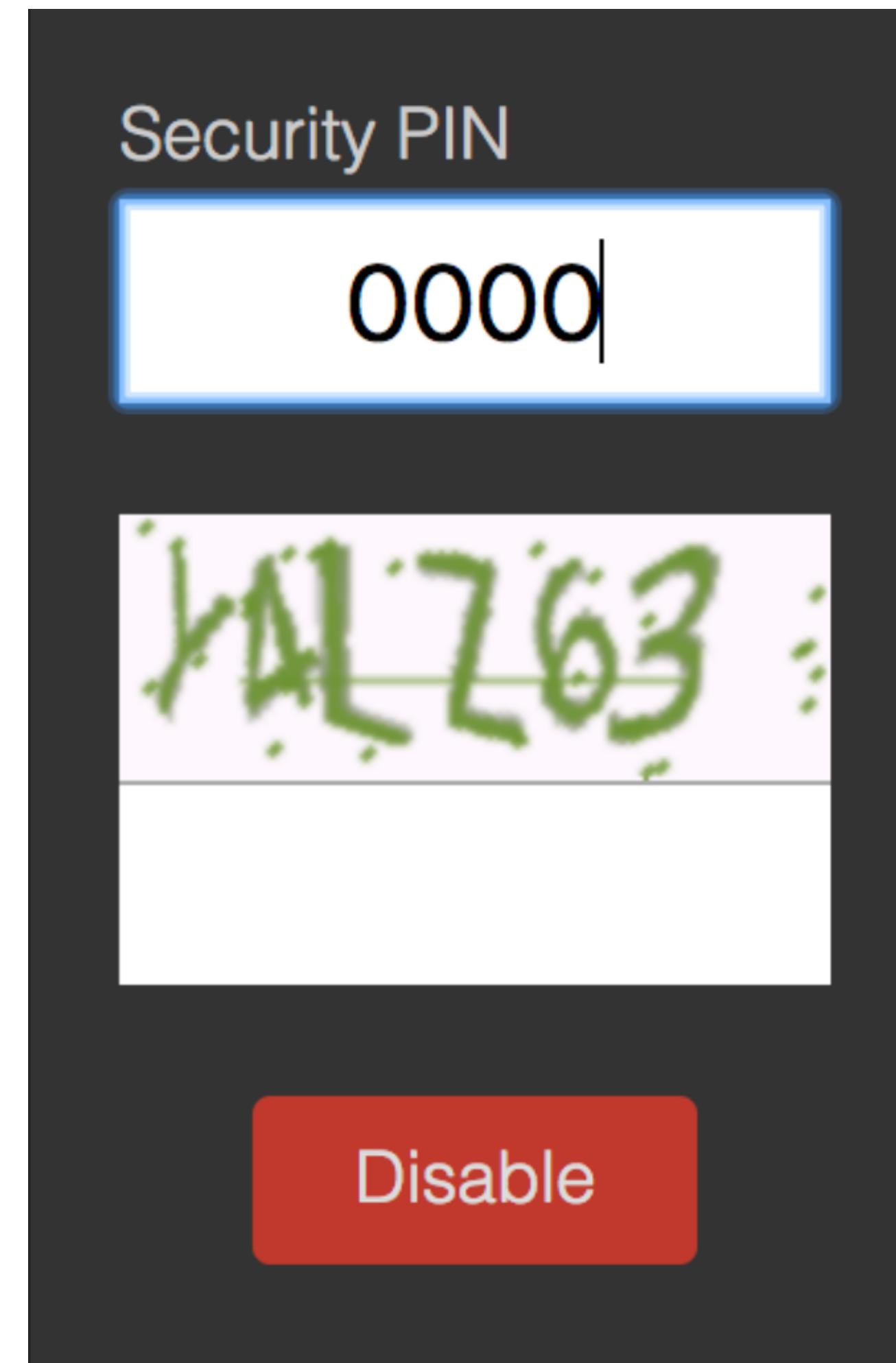
[blacklight.ai](https://blacklight.ai)

@blacklightai

# Captcha Challenge



# Captcha Challenge



# Captcha Challenge

1. Inspect the model



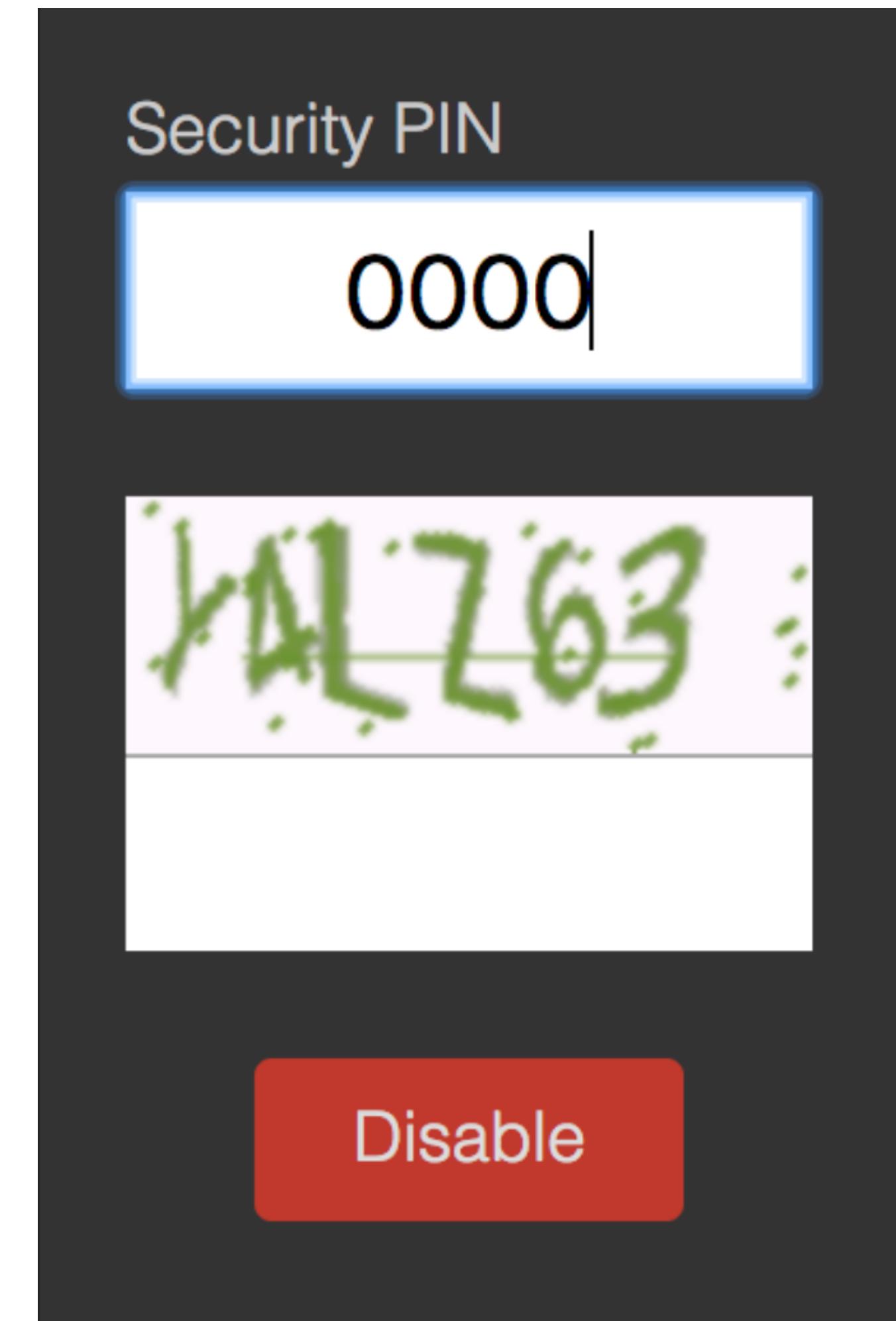
# Captcha Challenge

1. Inspect the model
2. Load the model



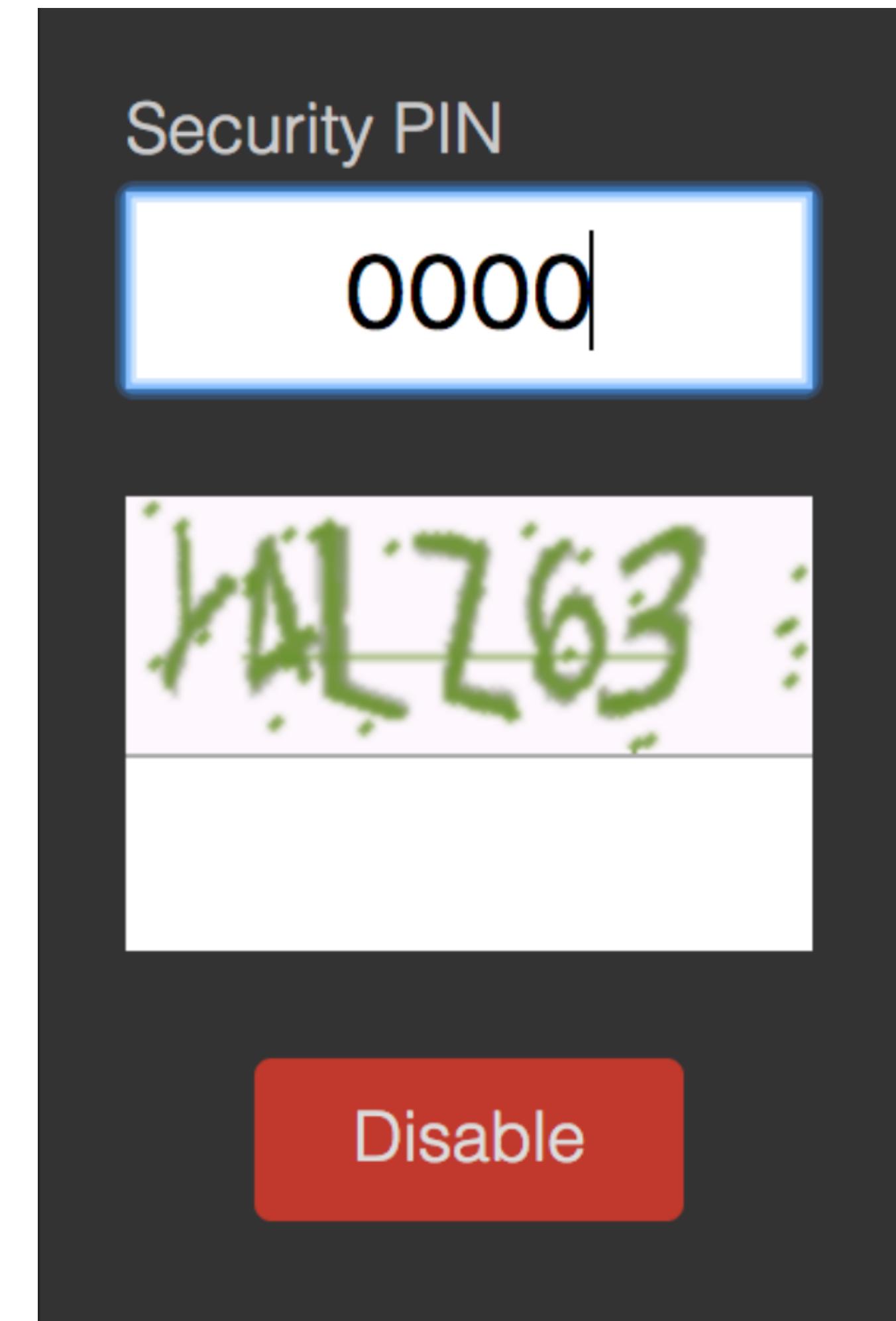
# Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:



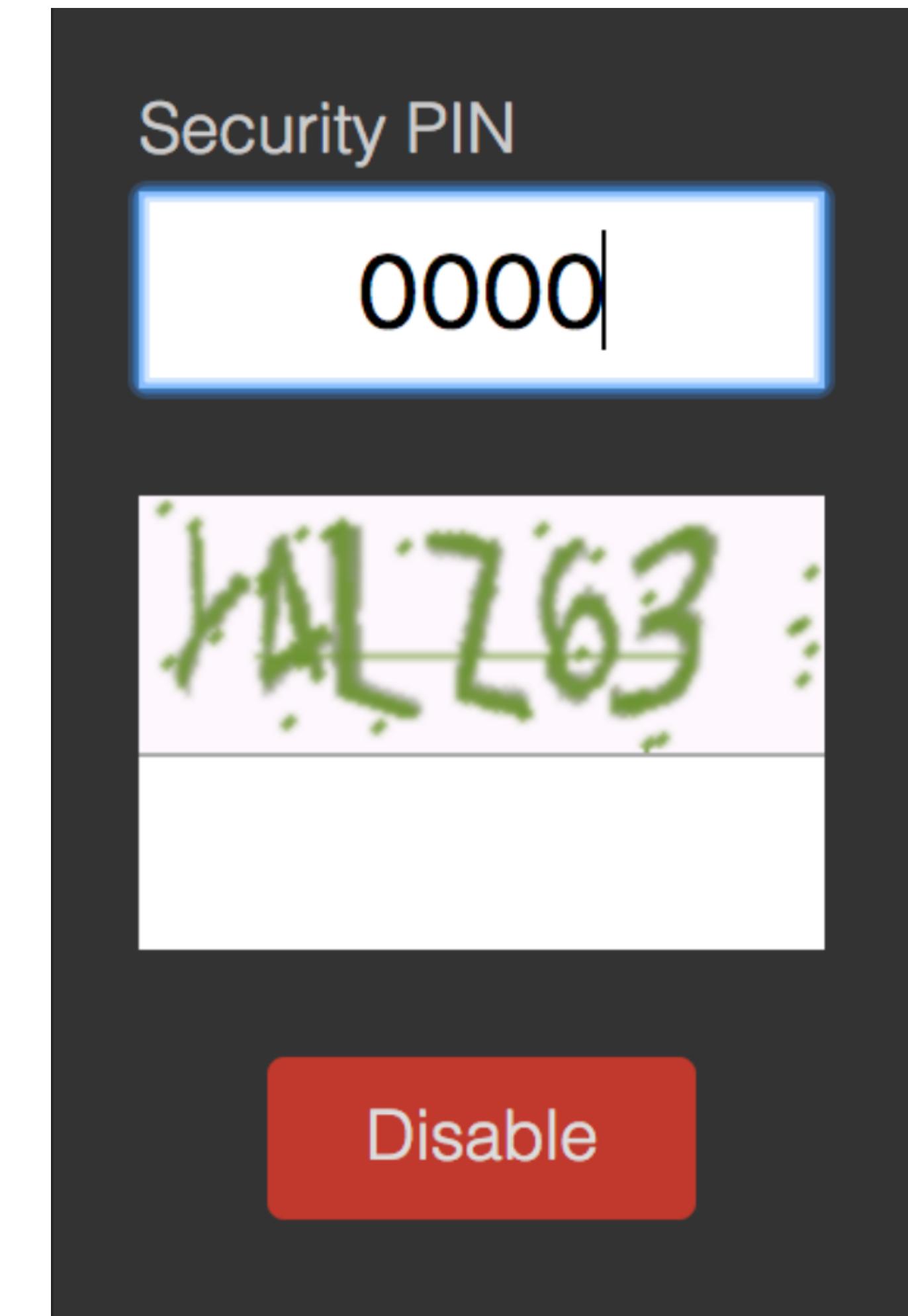
# Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
  - i. Open a cookie jar



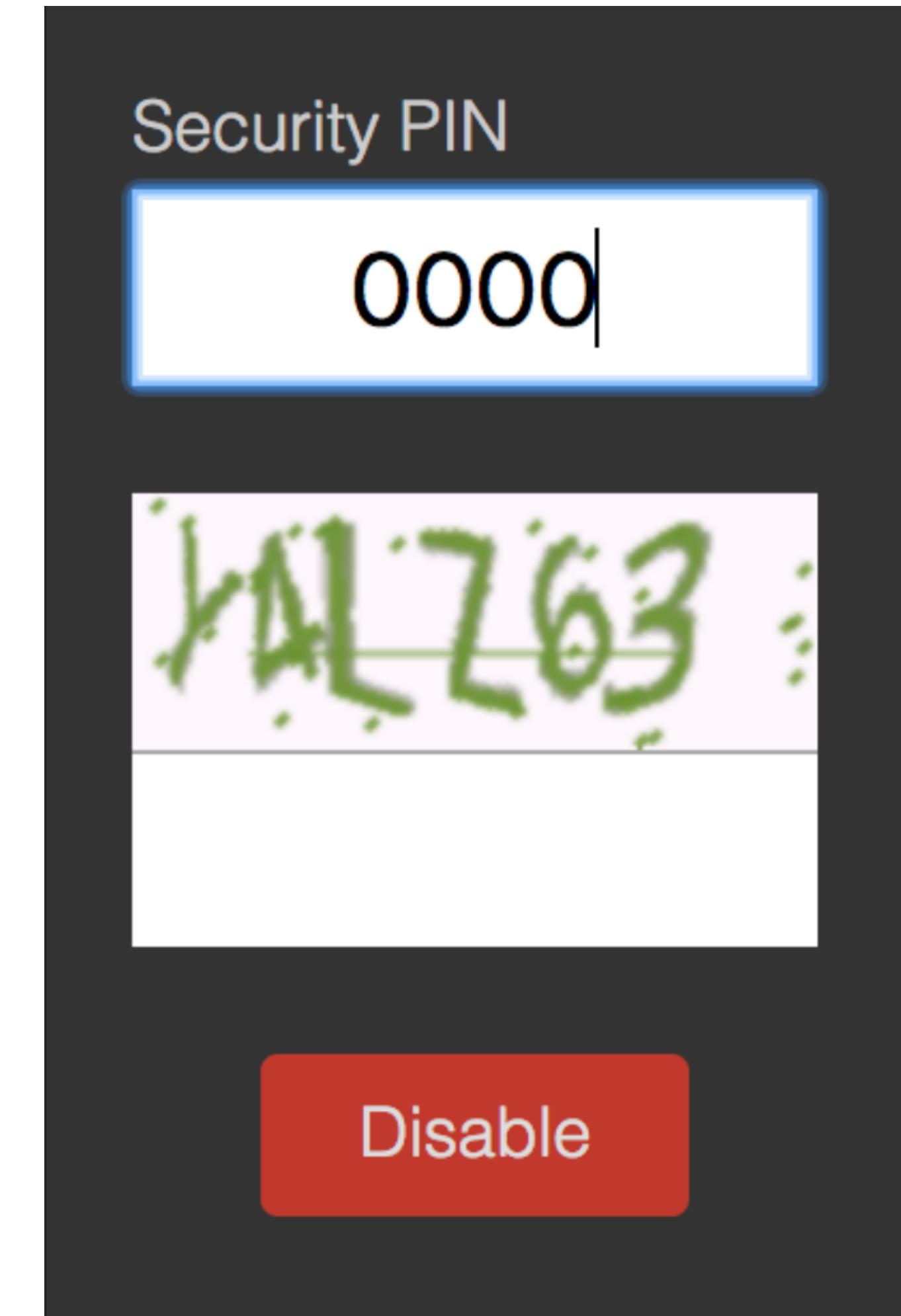
# Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
  - i. Open a cookie jar
  - ii. Get the CAPTCHA image



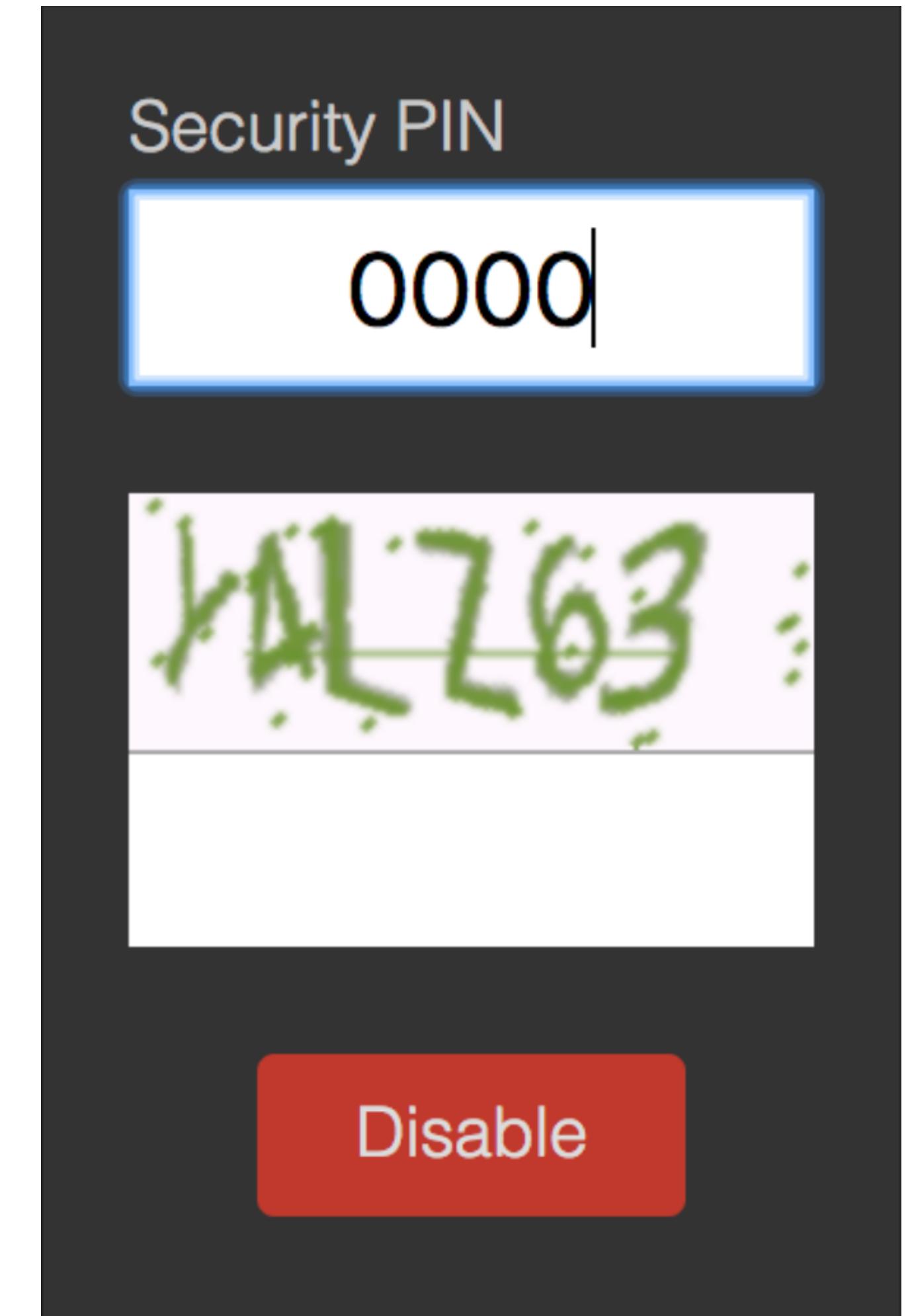
# Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
  - i. Open a cookie jar
  - ii. Get the CAPTCHA image
  - iii. Predict CAPTCHA using ML



# Captcha Challenge

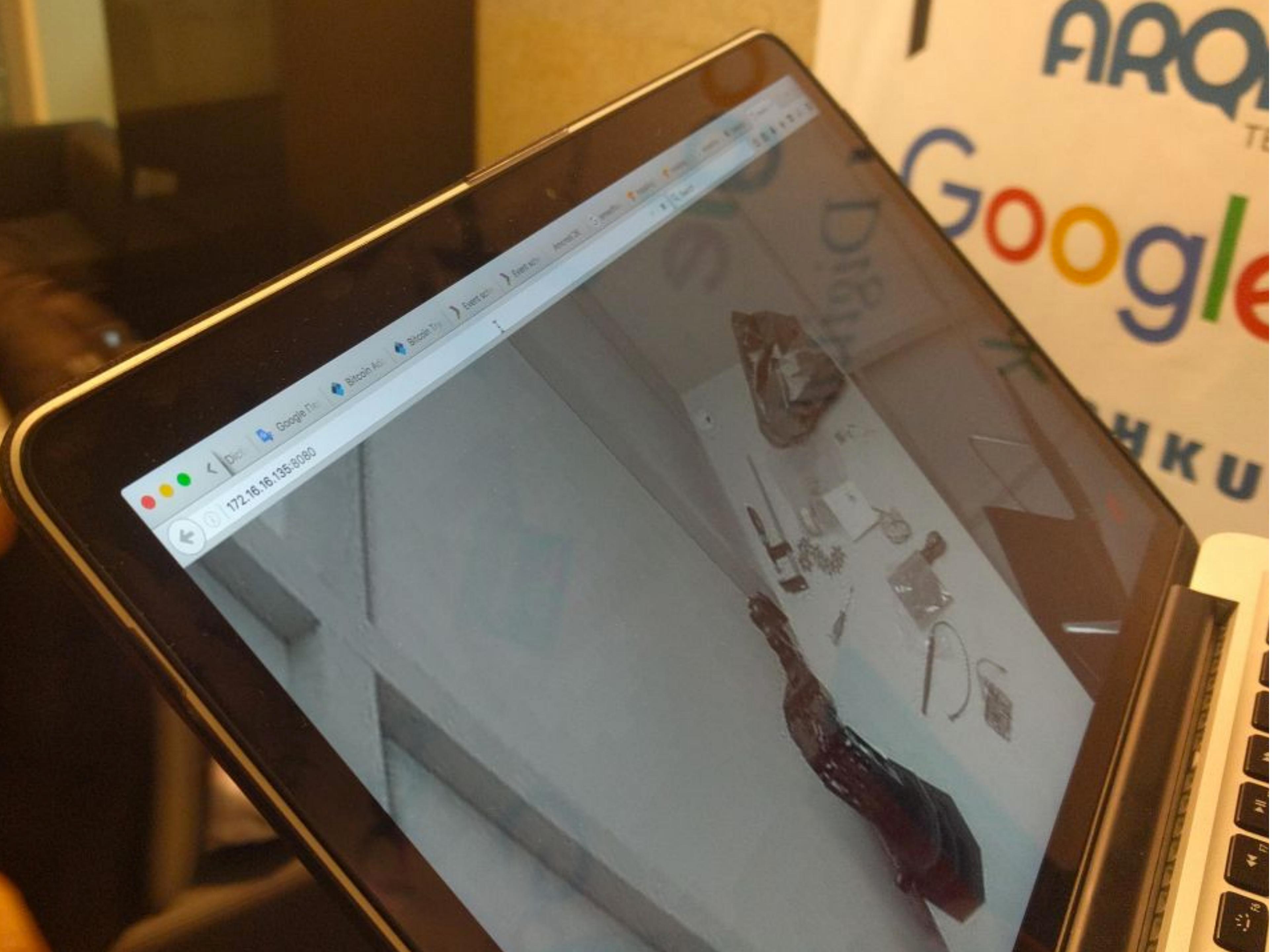
1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
  - i. Open a cookie jar
  - ii. Get the CAPTCHA image
  - iii. Predict CAPTCHA using ML
  - iv. Guess the PIN + CAPTCHA



# Captcha Challenge

1. Inspect the model
2. Load the model
3. Attempt logging in with the PIN:
  - i. Open a cookie jar
  - ii. Get the CAPTCHA image
  - iii. Predict CAPTCHA using ML
  - iv. Guess the PIN + CAPTCHA
    - a. if false CAPTCHA,  
fall back to (ii)



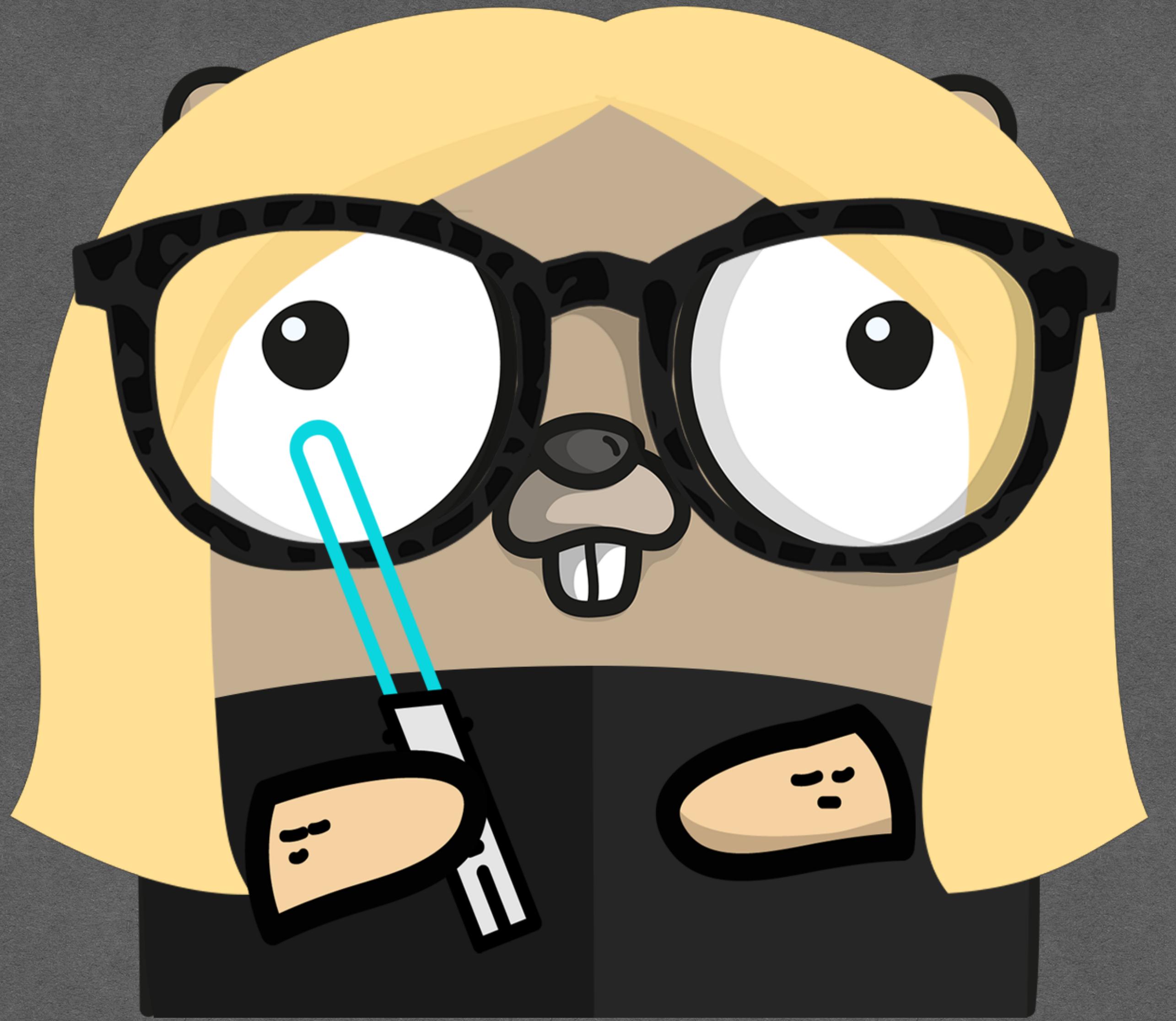


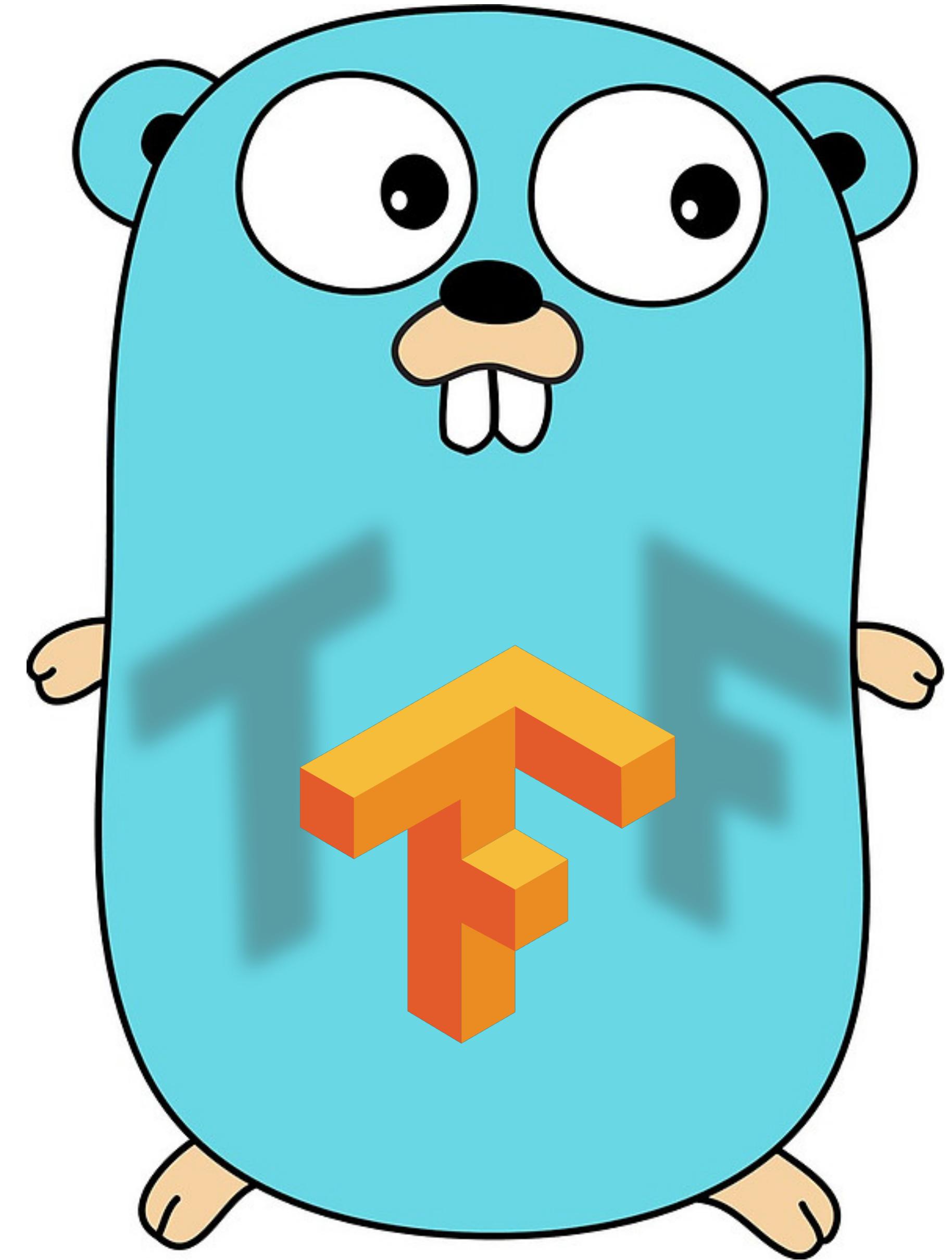
# Captcha Challenge

Read all about it at the December 28 2017  
Gophers Academy Advents Blog post

**<https://github.com/Pisush/break-captcha-tensorflow>**

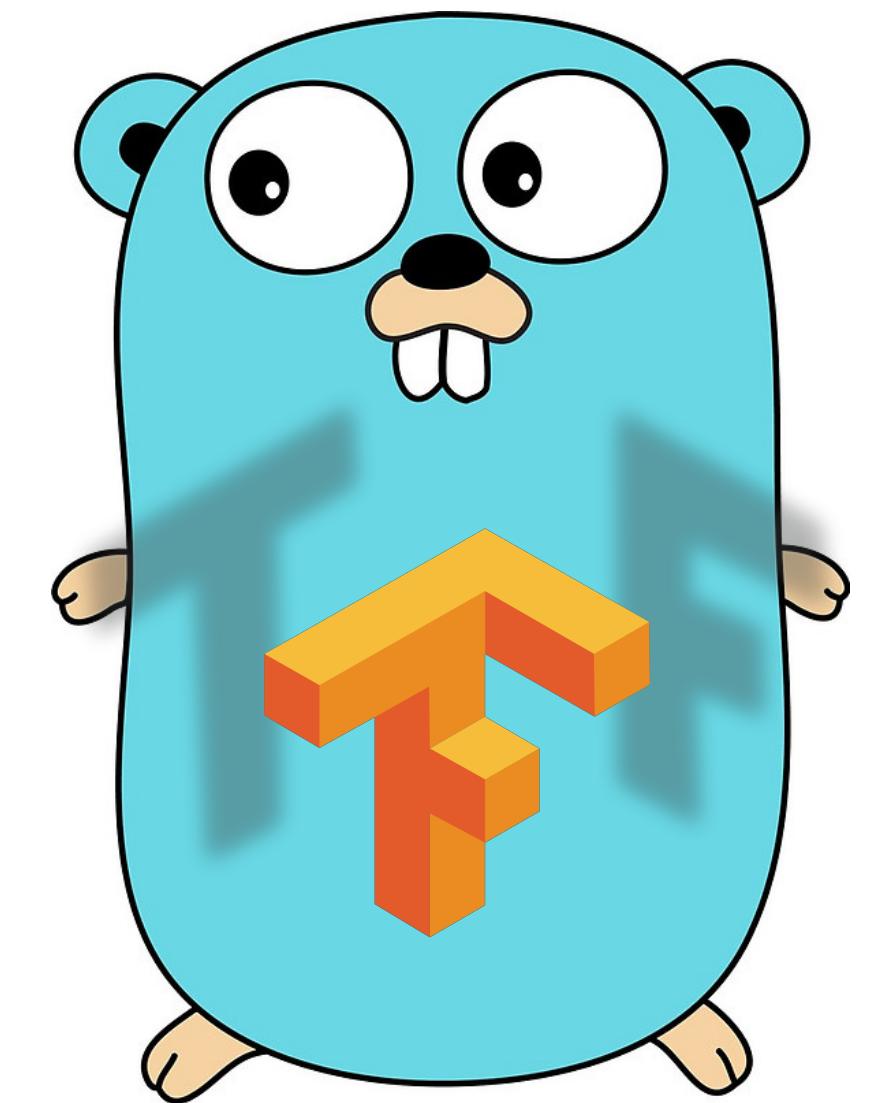
Using Machine Learning: Go + TensorFlow





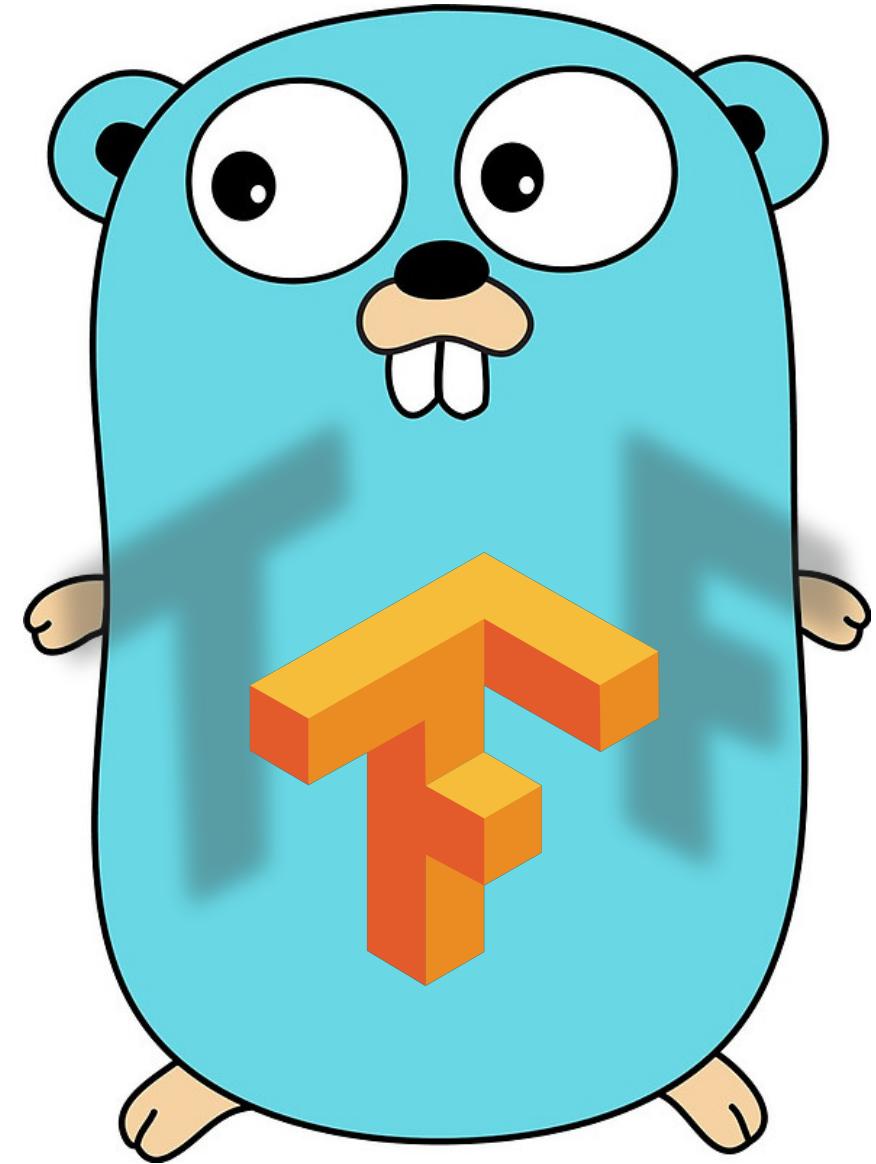
# How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



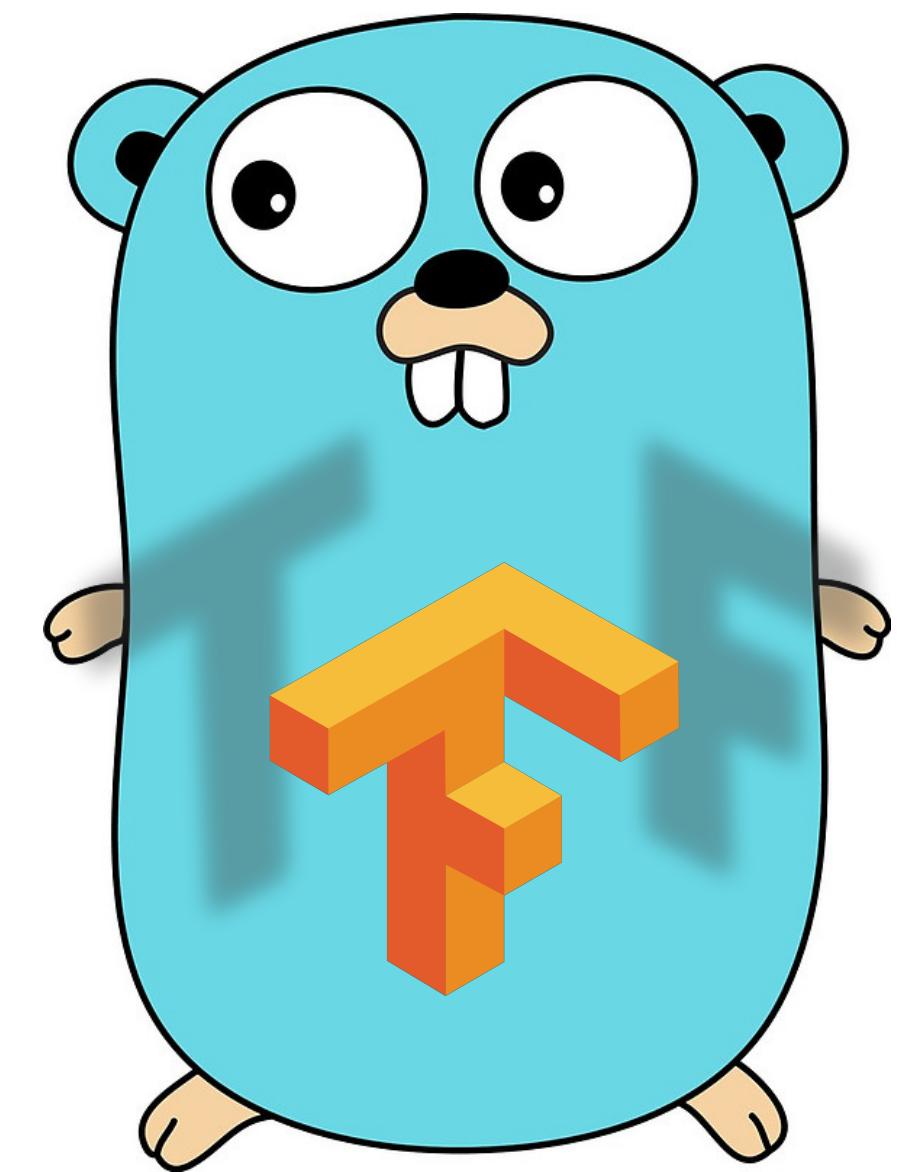
# How to ML

- 1. Define the problem**
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



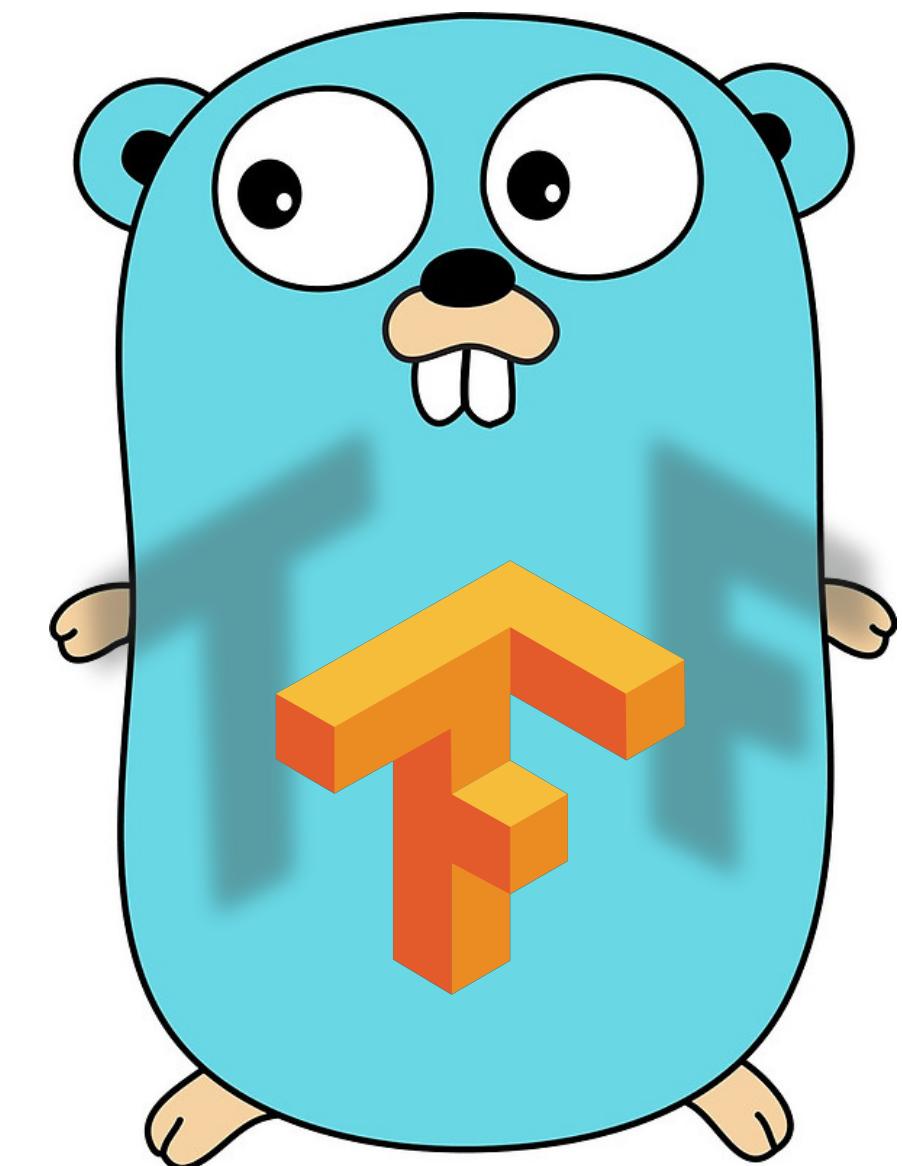
# How to ML

1. Define the problem
- 2. Gather data**
  - relevant to the task
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
8. Predict



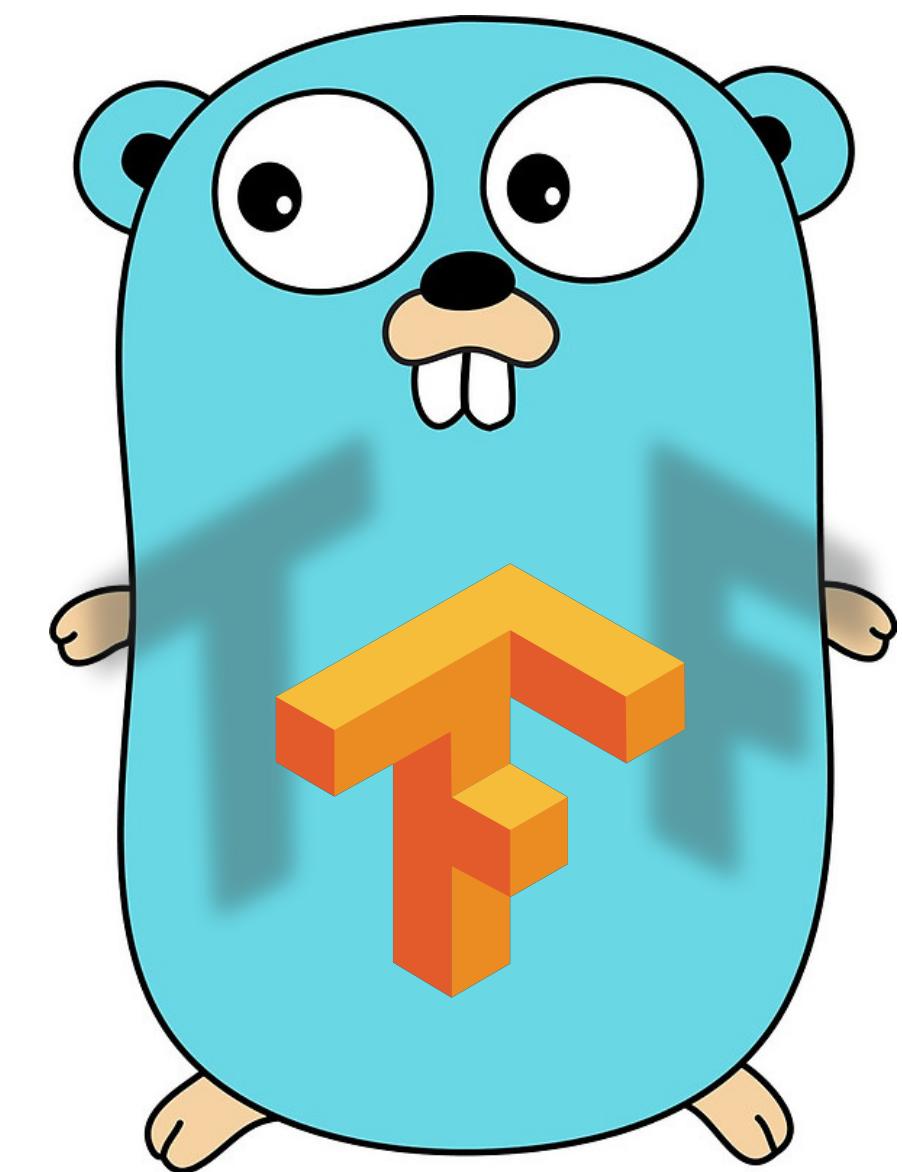
# How to ML

1. Define the problem
  2. Gather data
  3. **Prepare data**
  4. Choose a model
  5. Train the model
  6. Evaluate the model
  7. Tune the hyperparameters
  8. Predict
- [ ] clean and pre-process  
randomize  
split: train/test



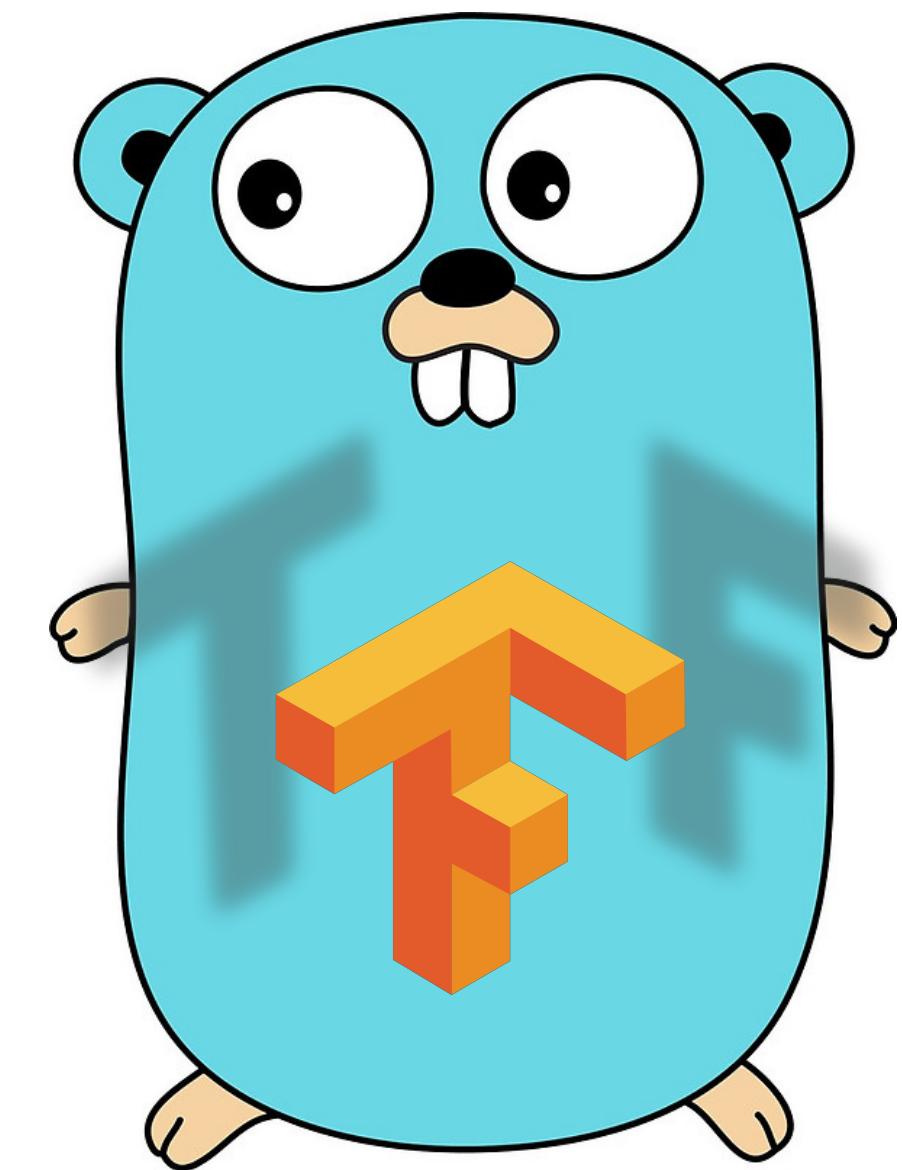
# How to ML

1. Define the problem
  2. Gather data
  3. **Prepare data**
  4. Choose a model
  5. Train the model
  6. Evaluate the model
  7. Tune the hyperparameters
  8. Predict
- [ ] clean and pre-process  
randomize  
split: train/test  
75/25



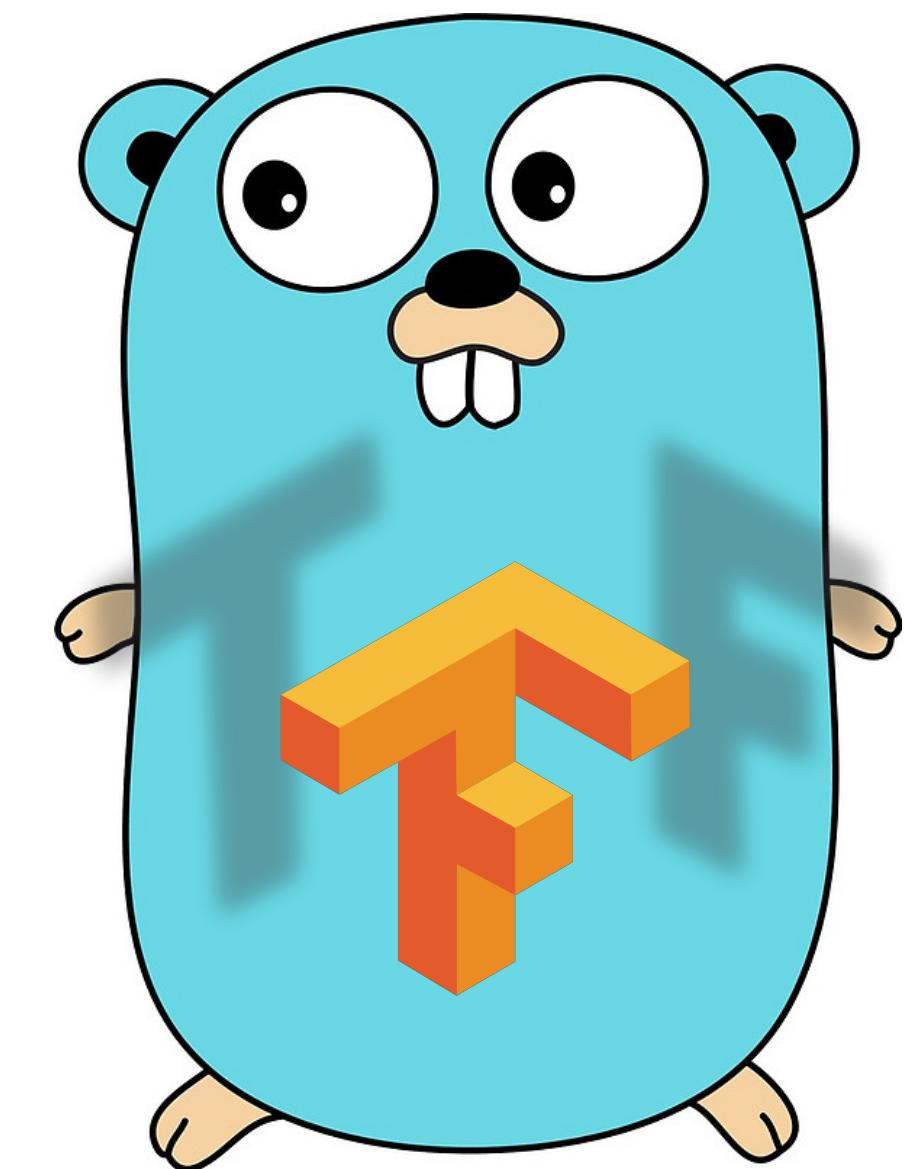
# How to ML

1. Define the problem
  2. Gather data
  3. Prepare data
  - 4. Choose a model**
  5. Train the model
  6. Evaluate the model
  7. Tune the hyperparameters
  8. Predict
- learning task  
input type  
possible number  
of categories



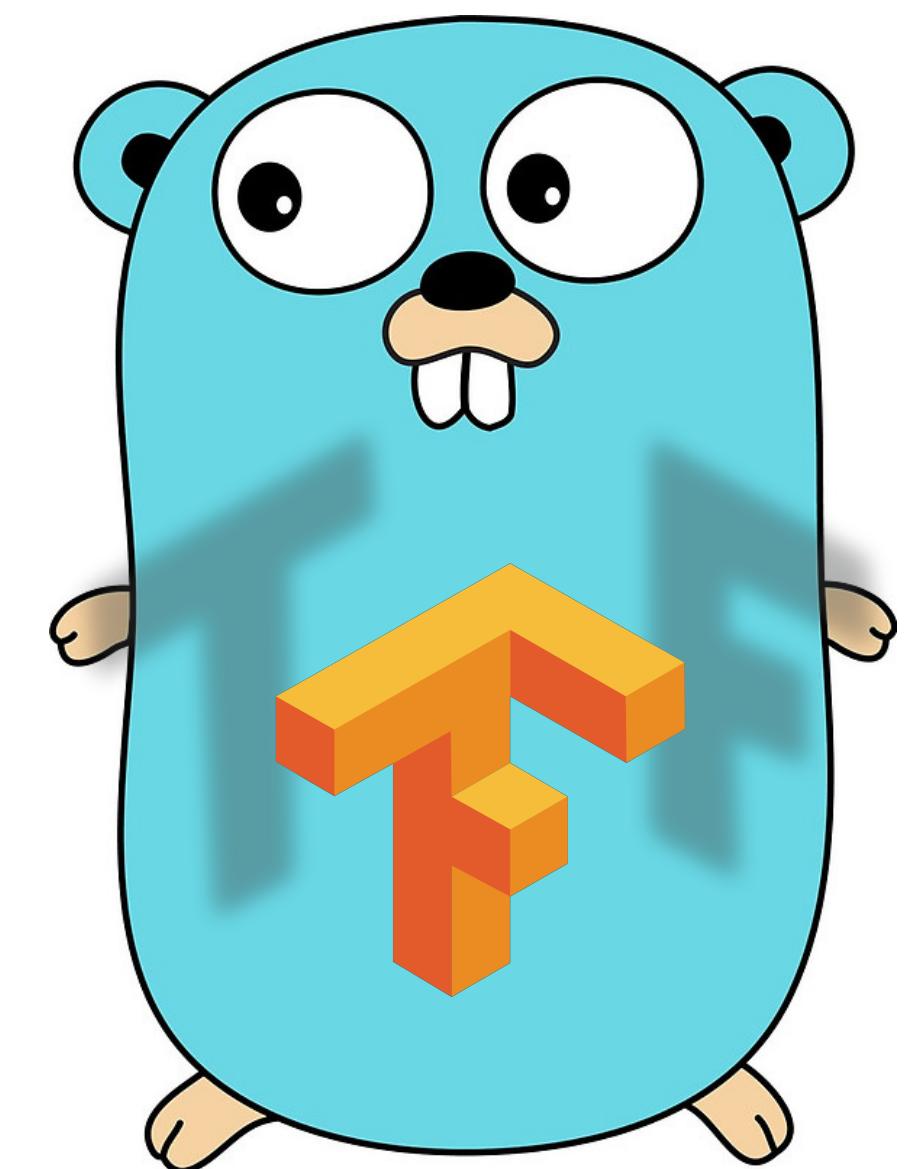
# How to ML

1. Define the problem
  2. Gather data
  3. Prepare data
  4. Choose a model
  - 5. Train the model**
  6. Evaluate the model
  7. Tune the hyperparameters
  8. Predict
- assign random values  
predict the train data  
adjust weights



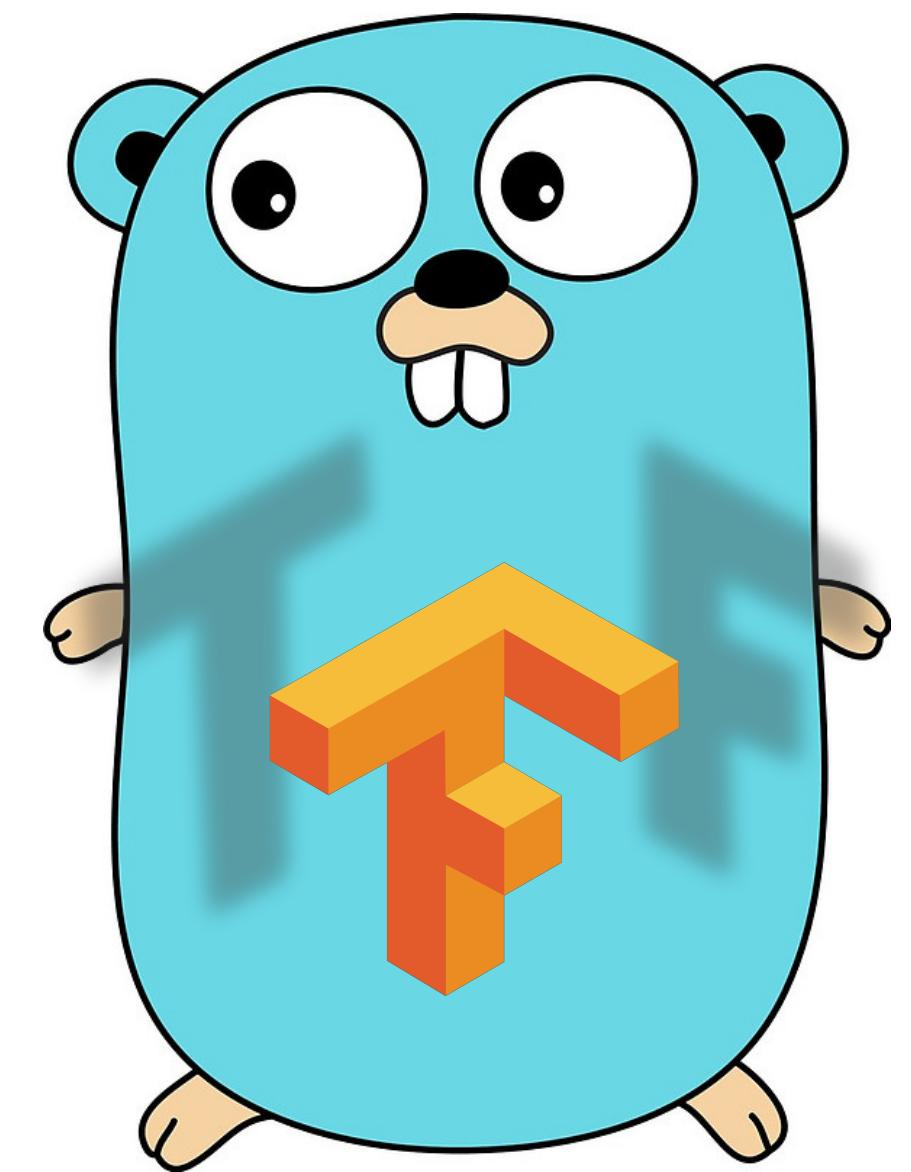
# How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
- 6. Evaluate the model**
  - check test data metrics
7. Tune the hyperparameters
8. Predict



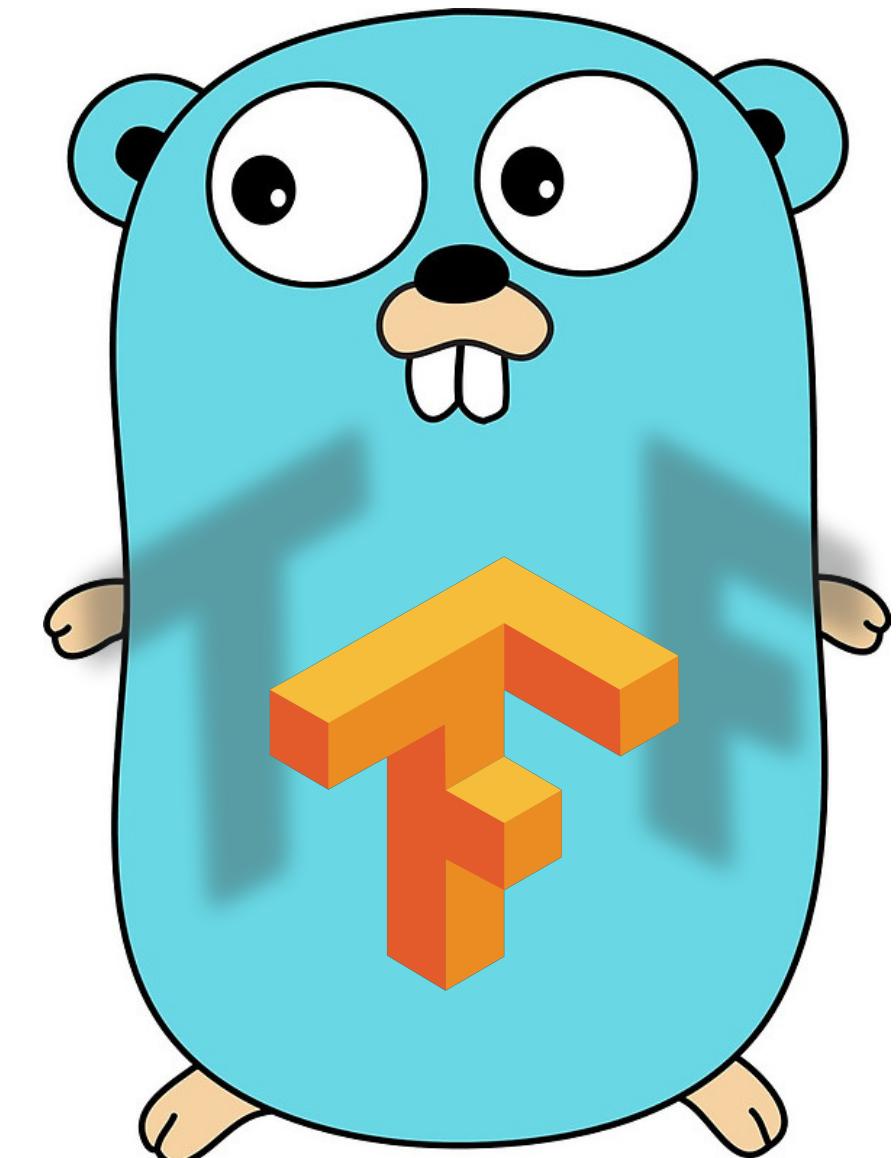
# How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
- 7. Tune the hyperparameters**
  - or, fine tune
8. Predict



# How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
7. Tune the hyperparameters
- 8. Predict**





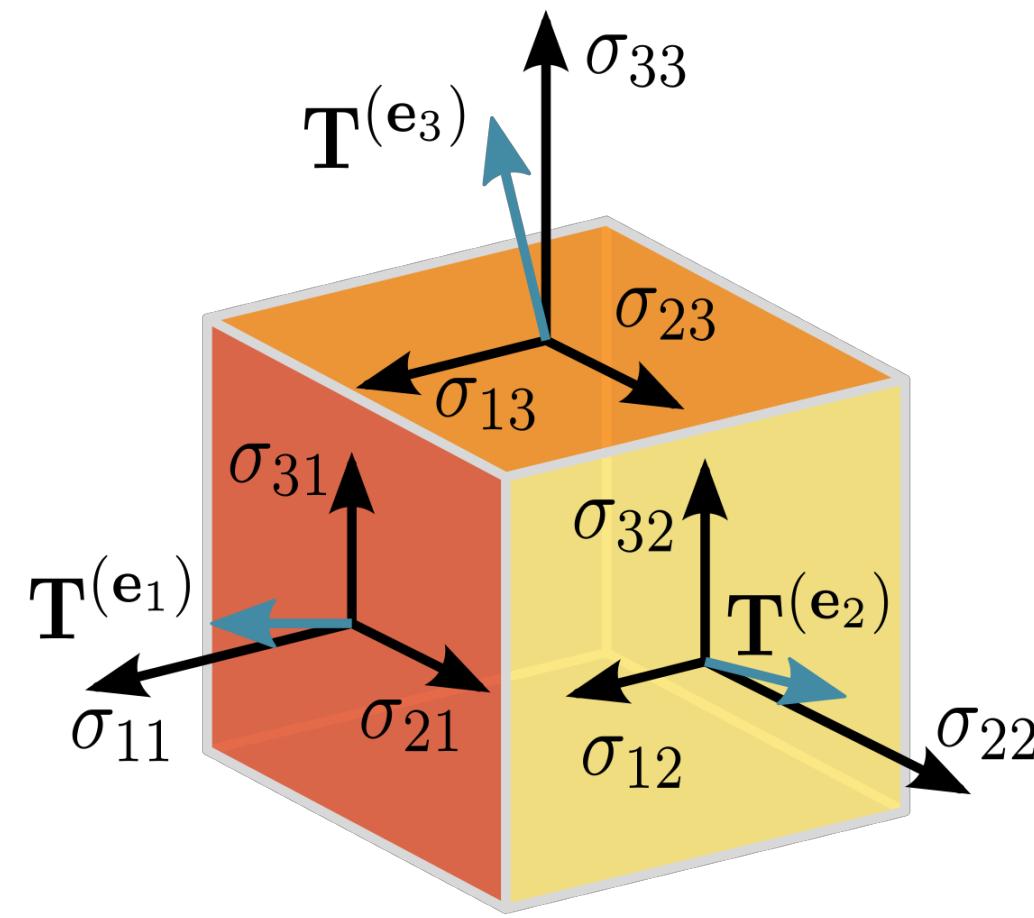
TensorFlow

**TensorFlow** is an open-source software  
for Machine Intelligence,  
used mainly for  
Machine Learning applications  
such as neural networks.



TensorFlow

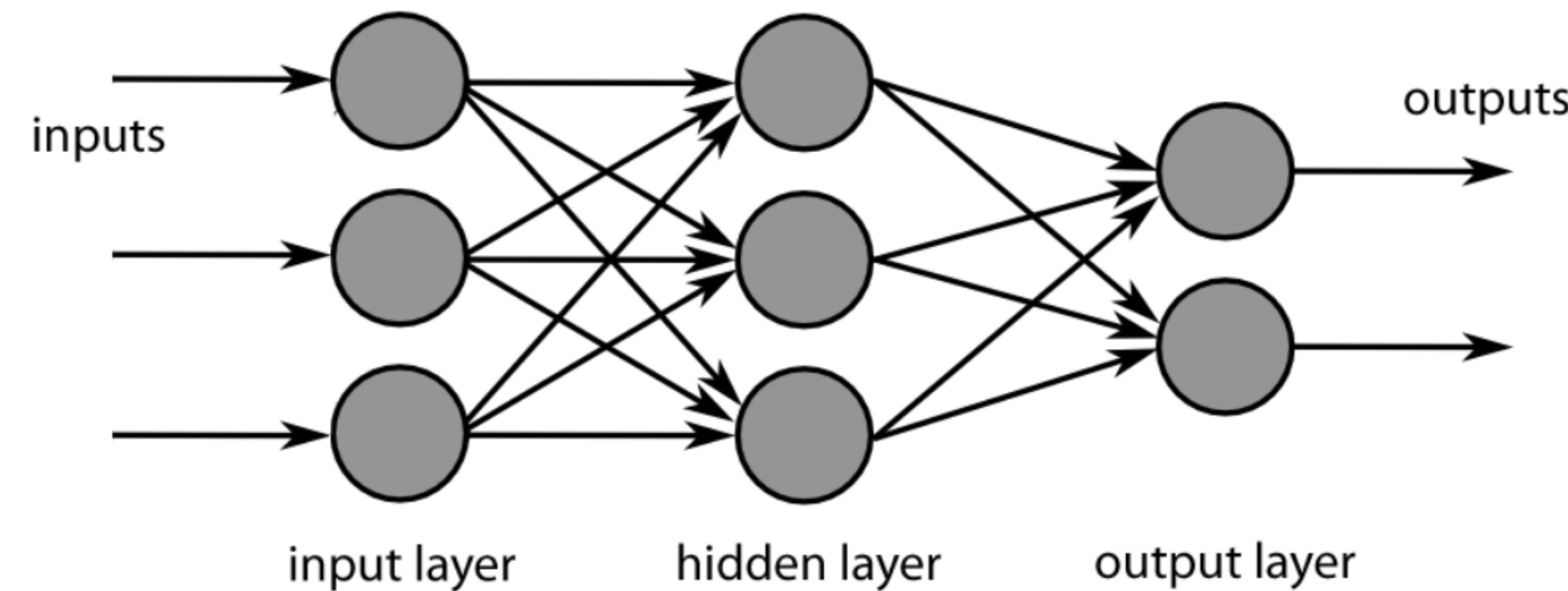
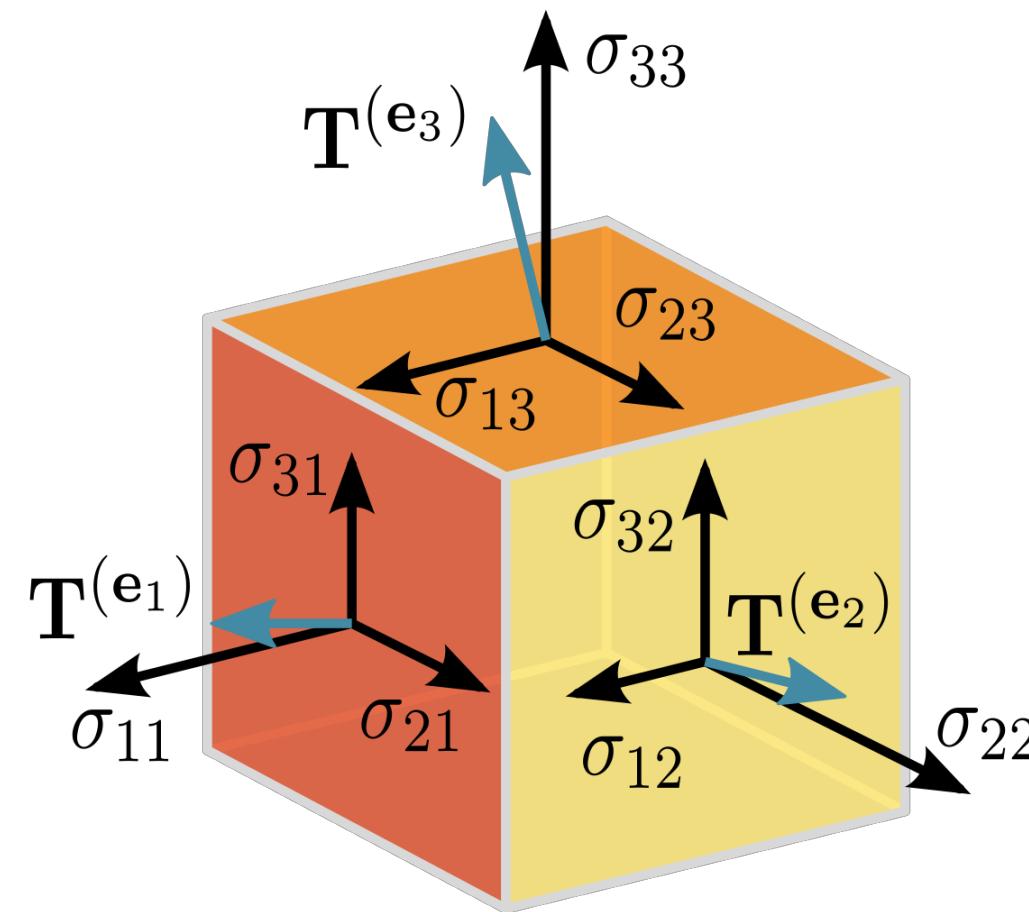
TensorFlow is an open-source software for Machine Intelligence, used mainly for machine learning applications such as neural networks.



A tensor is a generalization  
of vectors and matrices to  
potentially higher dimensions

1. data type
2. shape
  - number of dimensions
  - number of values / dimension

TensorFlow is an open-source software for Machine Intelligence, used mainly for machine learning applications such as neural networks.



A tensor is a generalization of vectors and matrices to potentially higher dimensions

1. data type
2. shape
  - number of dimensions
  - number of values / dimension

The flow part comes to describe:  
- the graph (model) is a set of nodes (operations)  
- the data (tensors) "flows" through those nodes, undergoing mathematical manipulation

You can look at, and evaluate, any node of the graph

# TensorFlow

- Community driven
- Becoming friendly for developers
  - AutoML: automates ML models design
  - TF Hub: repo for modules
  - Black-box tools built on top of TF

# TensorFlow

- Community driven
- Becoming friendly for developers
  - **AutoML: automates ML models design**
  - TF Hub: repo for modules
  - Black-box tools built on top of TF

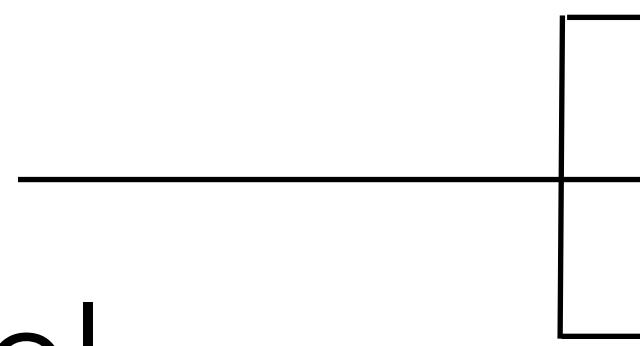
# How to ML

1. Define the problem
2. Gather data
3. Prepare data
4. Choose a model
5. Train the model
6. Evaluate the model
- 7. Tune the hyperparameters**
  - or, fine tune
8. Predict

# TensorFlow

- Community driven
- Becoming friendly for developers
  - AutoML: automates ML models design
  - **TF Hub: repo for modules**
  - Black-box tools built on top of TF

# How to ML

1. Define the problem
  2. Gather data
  3. Prepare data
  4. Choose a model
  5. **Train the model**
  6. Evaluate the model
  7. Tune the hyperparameters
  8. Predict
- 
- assign random values
  - predict the train data
  - adjust weights

# TensorFlow

- Community driven
- Becoming friendly for developers
  - AutoML: automates ML models design
  - TF Hub: repo for modules
  - **Black-box tools built on top of TF**

# Supported Languages

## TF APIs

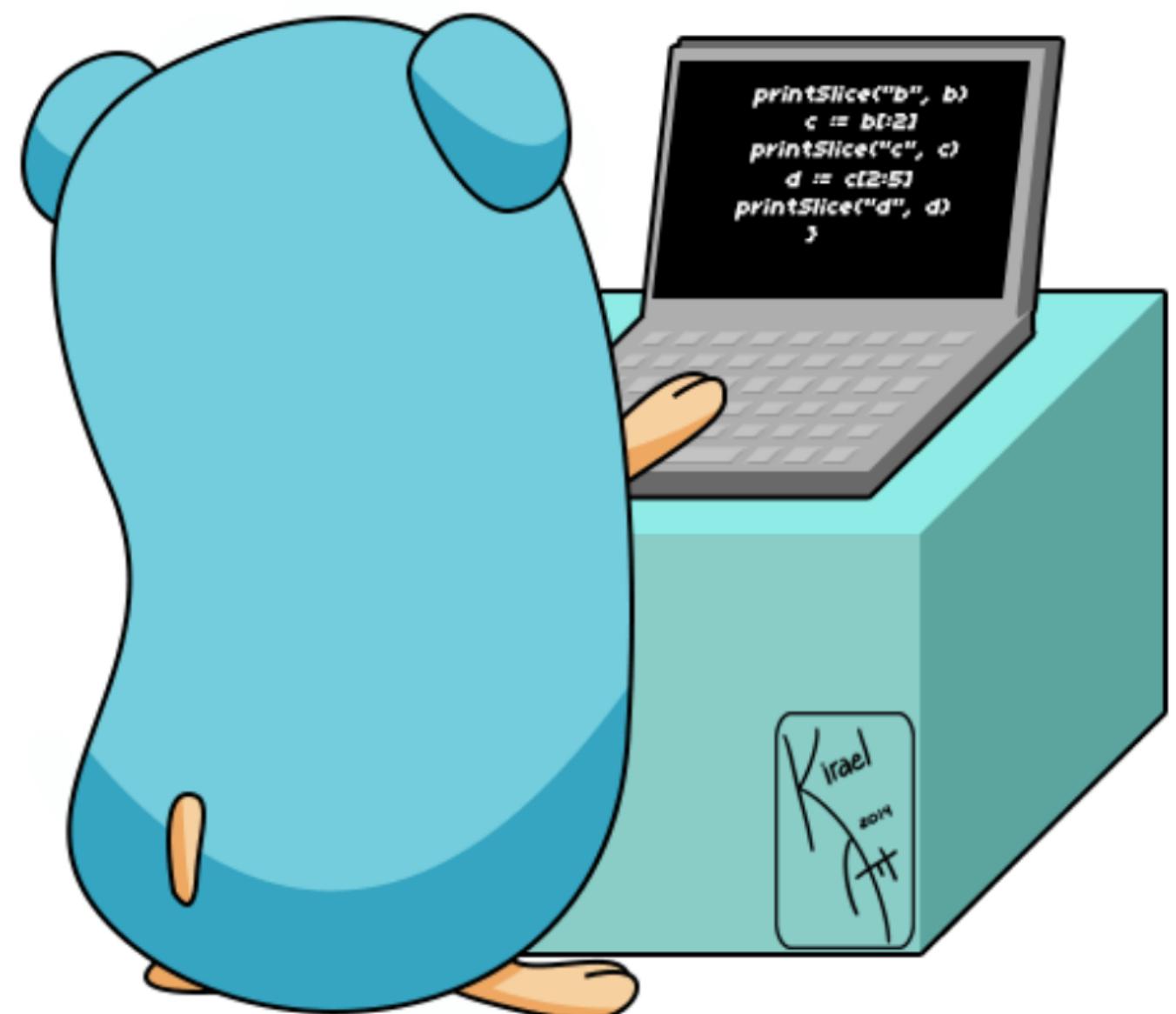
- Python
- C++
- Java
- Go
- JavaScript
- Swift

## TF Bindings

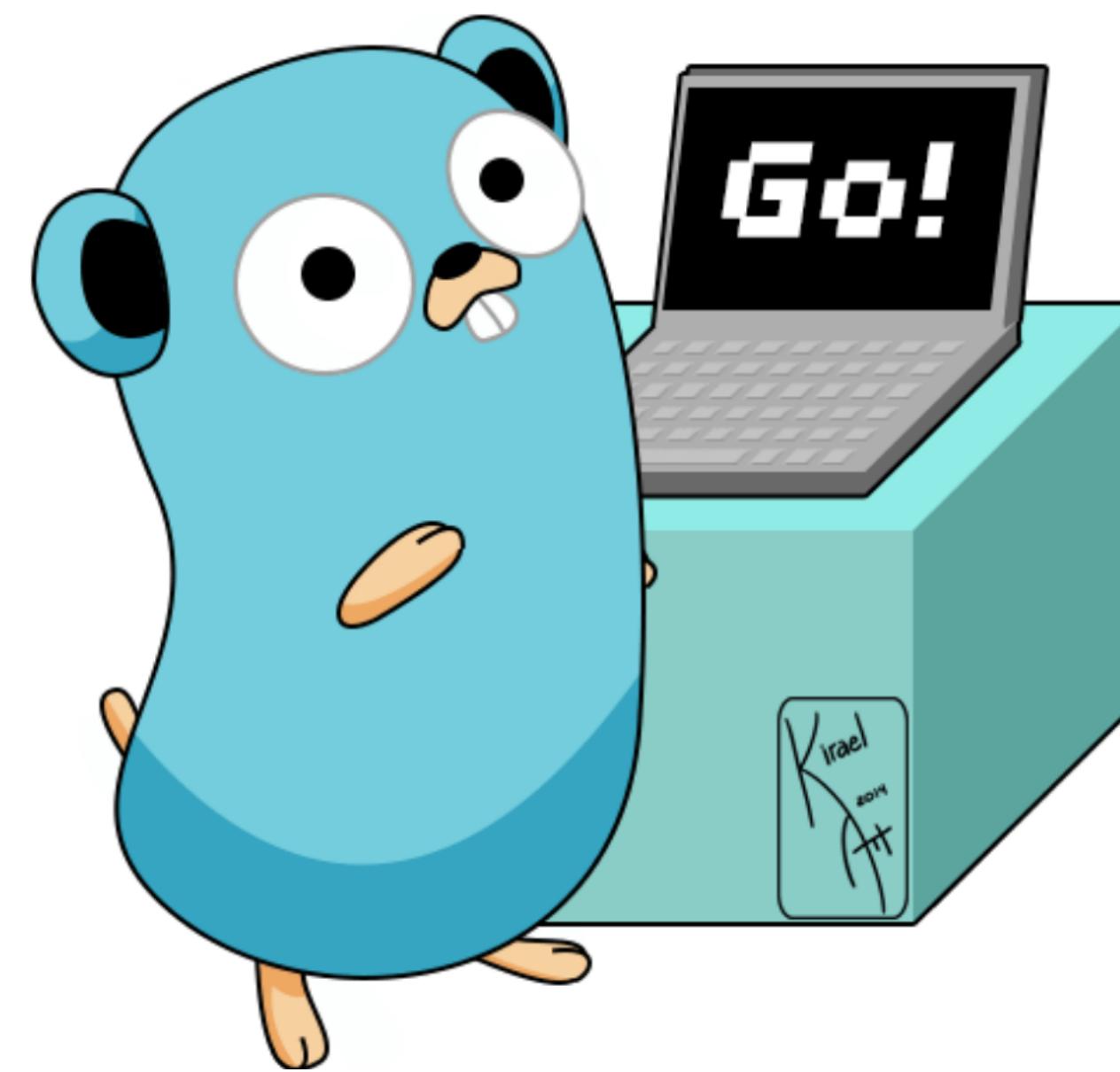
- C#
- Haskell
- Julia
- Ruby
- Rust
- Scala
-

# Go APIs for TF

Train models

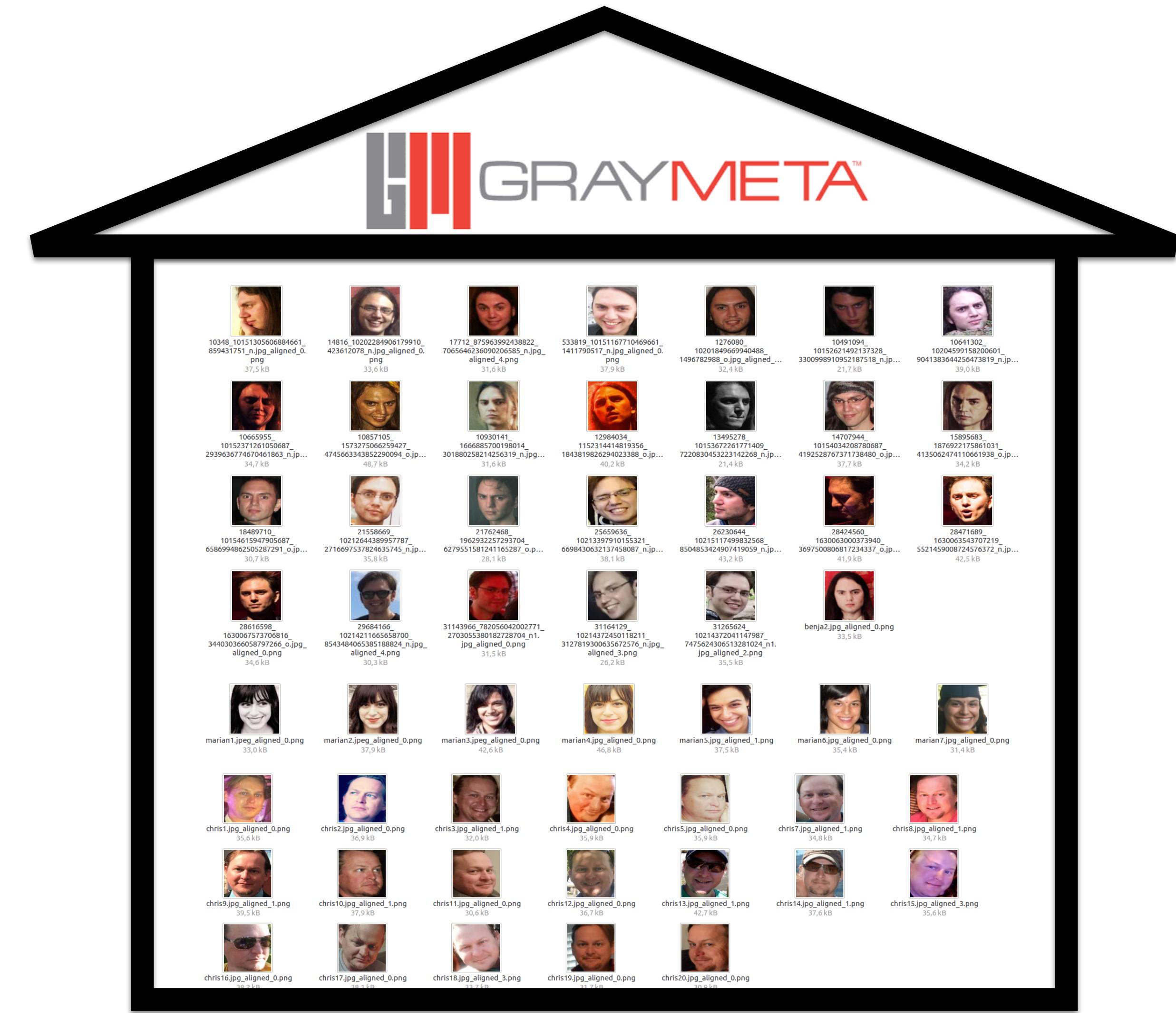


Serve models



# #2

# In-house ML



# In-house ML

Let's recognise faces!

Requirements:

- Thousands of categories
- Increasing number of categories
- Many categories can be added any time
- Learn a new category from 2-3 example
- Real-time classifier retraining
- Fast predictions

# In-house ML

The steps we took:

# In-house ML

The steps we took:

1. Gather data

# In-house ML

The steps we took:

1. Gather data
  - Mix of open and private data sets
2. Prepare data

# In-house ML

The steps we took:

1. Gather data
  - Mix of open and private data sets
2. Prepare data
3. Choose a model
  - FaceNet / OpenFace / custom and dlib-based

# In-house ML

The steps we took:

1. Gather data
  - Mix of open and private data sets
2. Prepare data
3. Choose a model
  - FaceNet / OpenFace / ~~custom and dlib-based~~

# In-house ML

The steps we took:

1. Gather data
  - Mix of open and private data sets
2. Prepare data
3. Choose a model
  - FaceNet / OpenFace / ~~custom and dlib-based~~
  - Built-in classifier on top of the output vectors

# In-house ML

The steps we took:

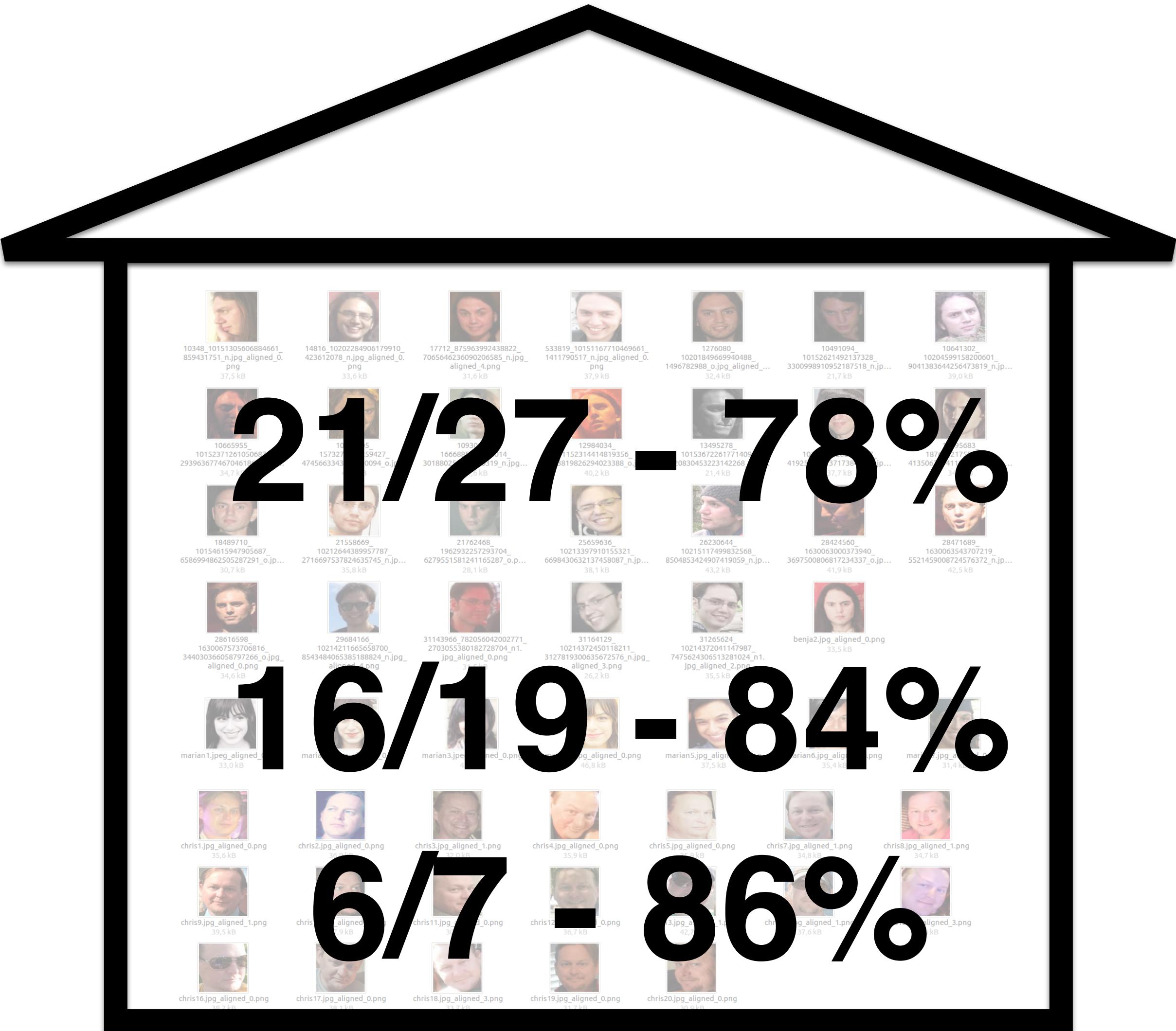
1. Gather data
  - Mix of open and private data sets
2. Prepare data
3. Choose a model
  - FaceNet / OpenFace / ~~custom and dlib-based~~
  - Built-in classifier on top of the output vectors
4. Train the model
  - Unique per client
  - Write Go code to run the model, integrate it to the engine

# In-house ML

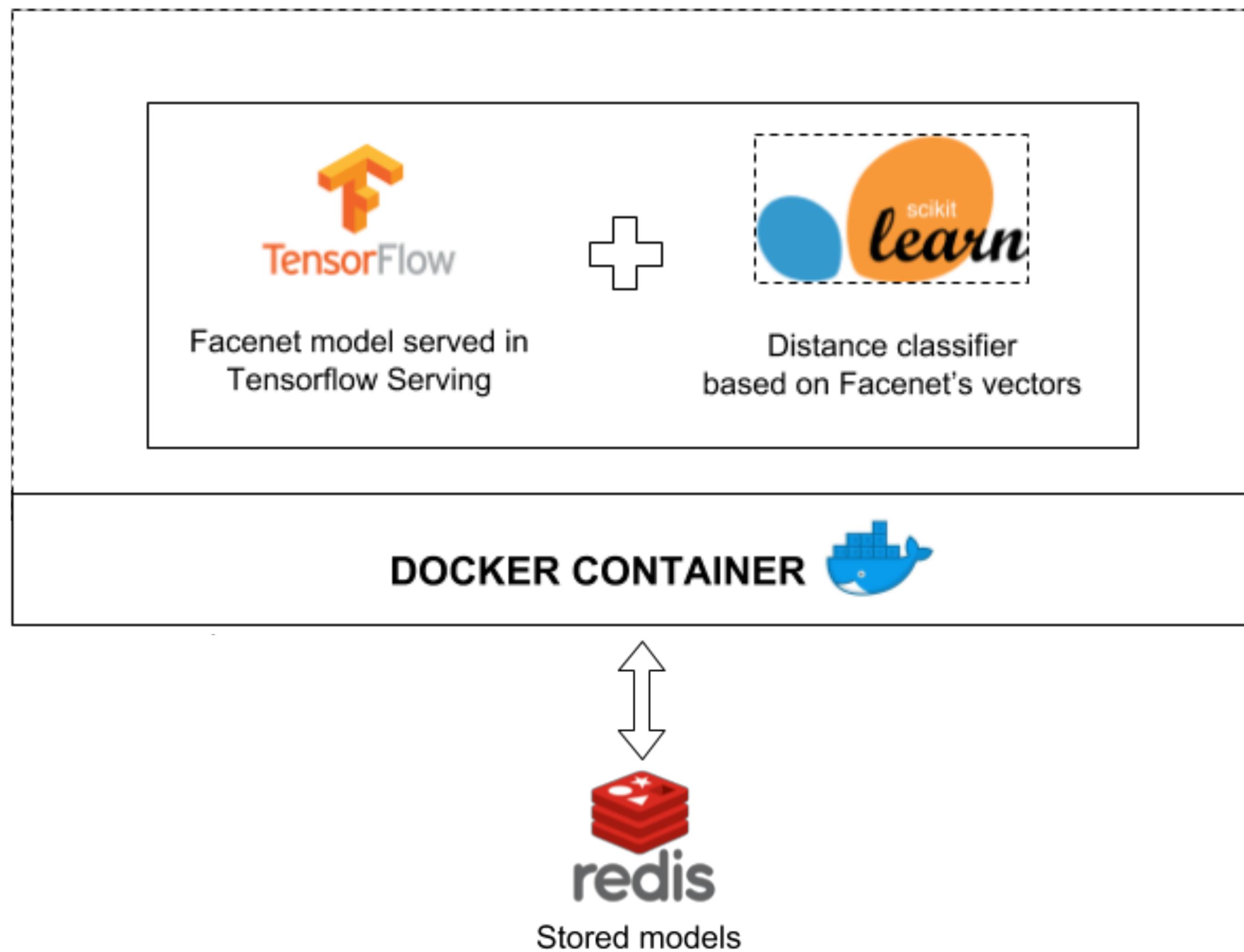
The steps we took:

1. Gather data
  - Mix of open and private data sets
2. Prepare data
3. Choose a model
  - FaceNet / OpenFace / custom and dlib based
  - Built-in classifier on top of the output vectors
4. Train the model
  - Unique per client
  - Write Go code to run the model, integrate it to the engine
5. Evaluate the model
6. Tune the parameters
7. Predict

# In-house ML



# In-house ML

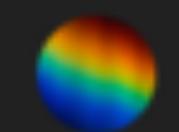


# PhotoPrism

#3



Golang Berlin



PhotoPrism



Chicago



Photos



Places



Calendar



Tags



Favorites



Bookmarks



Filters



Albums



Import



Export



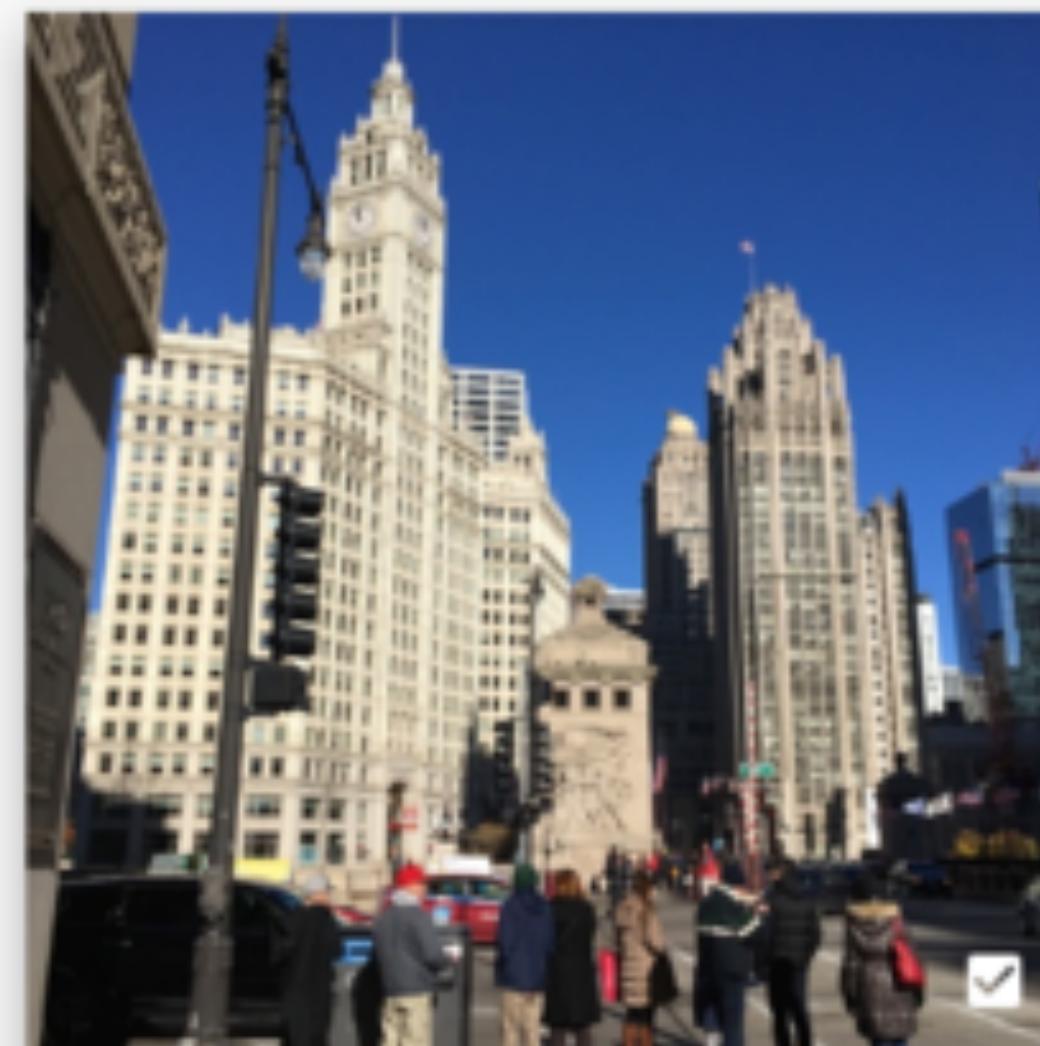
Settings

USA

iPhone SE

Details

Recently imported



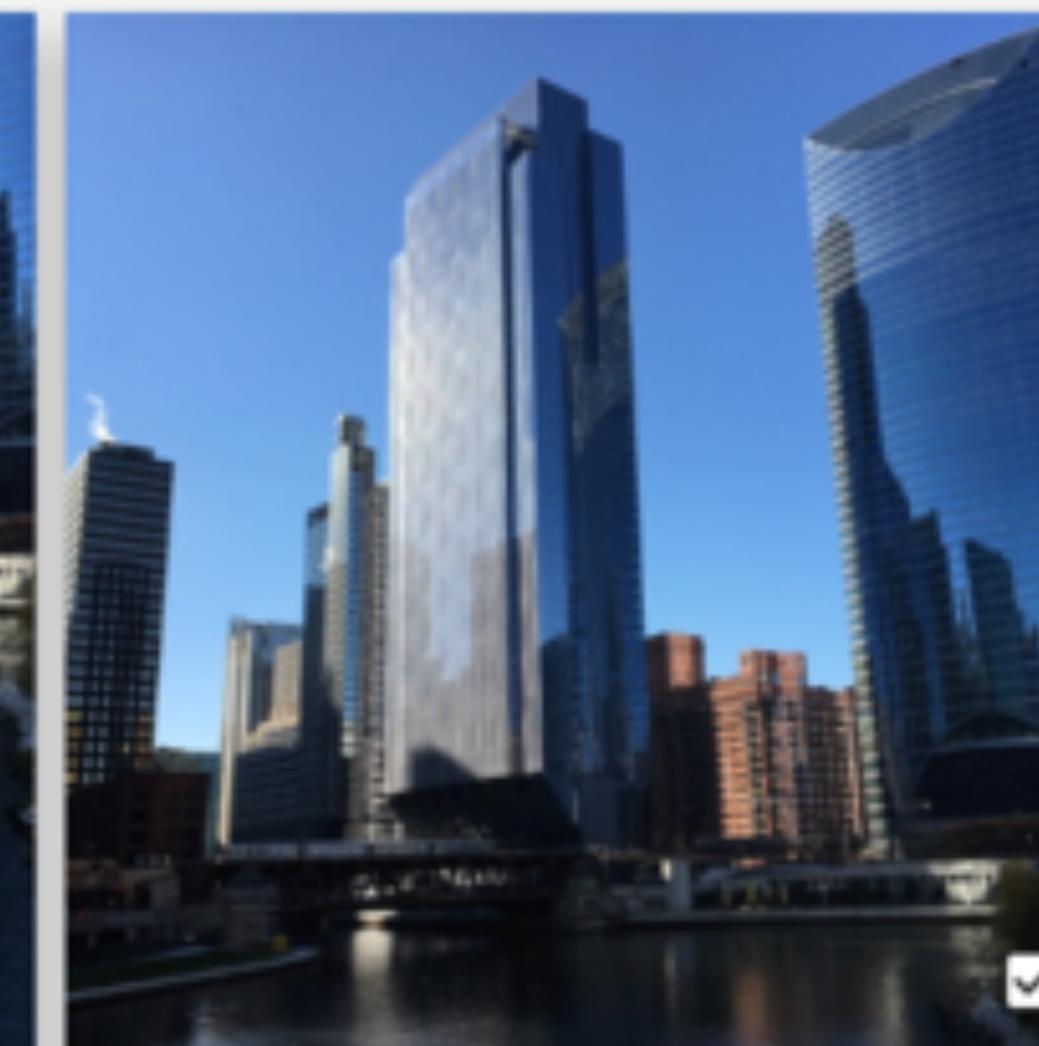
London Guarantee Building / Chicago / 2016

21/11/2016 12:01:45  
iPhone SE  
Chicago, Illinois, USA



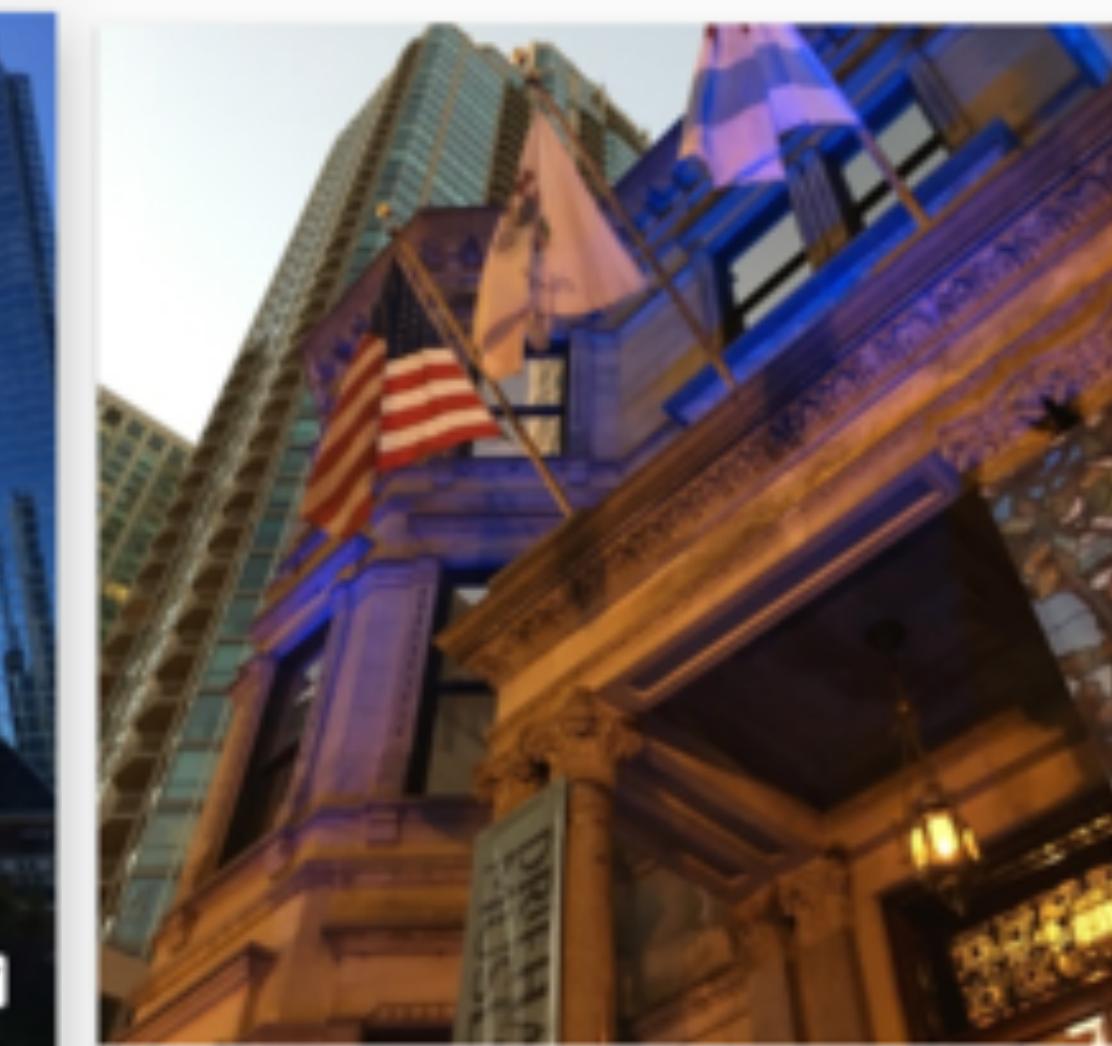
View On Chicago River / Chicago / 2016

21/11/2016 11:11:25  
iPhone SE  
Chicago, Illinois, USA



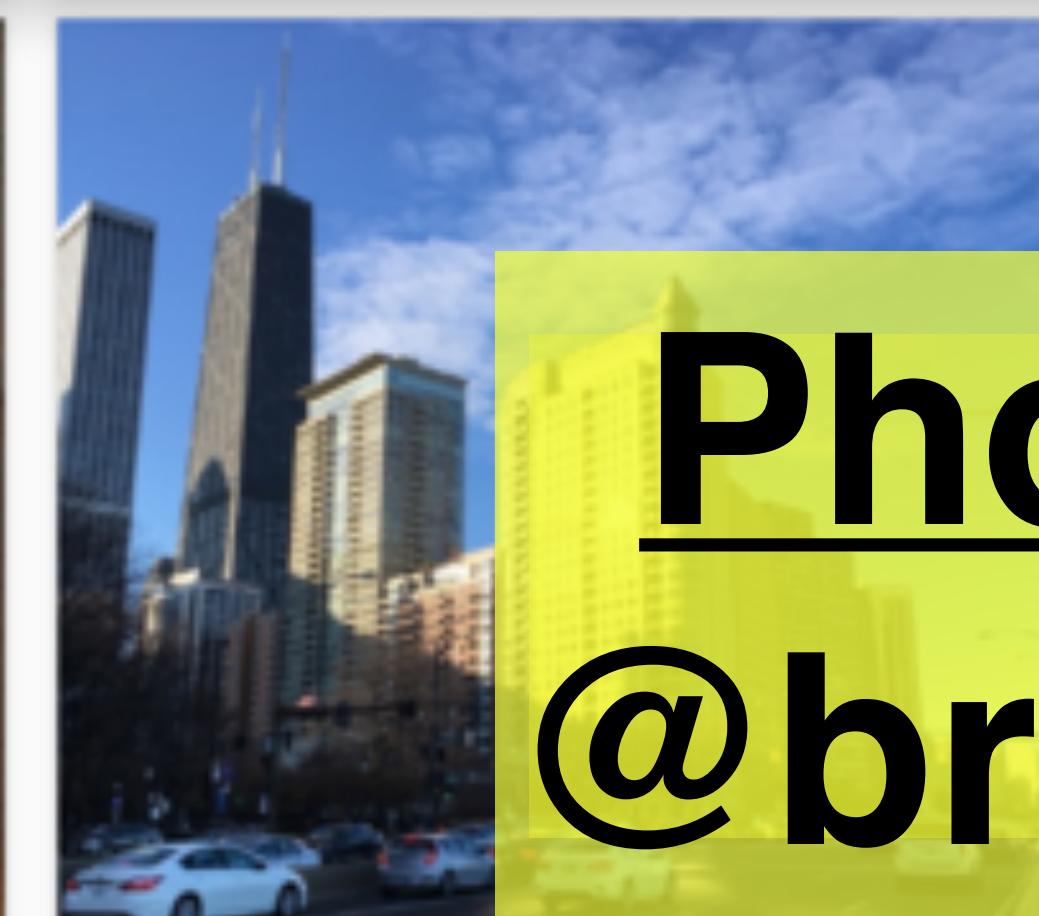
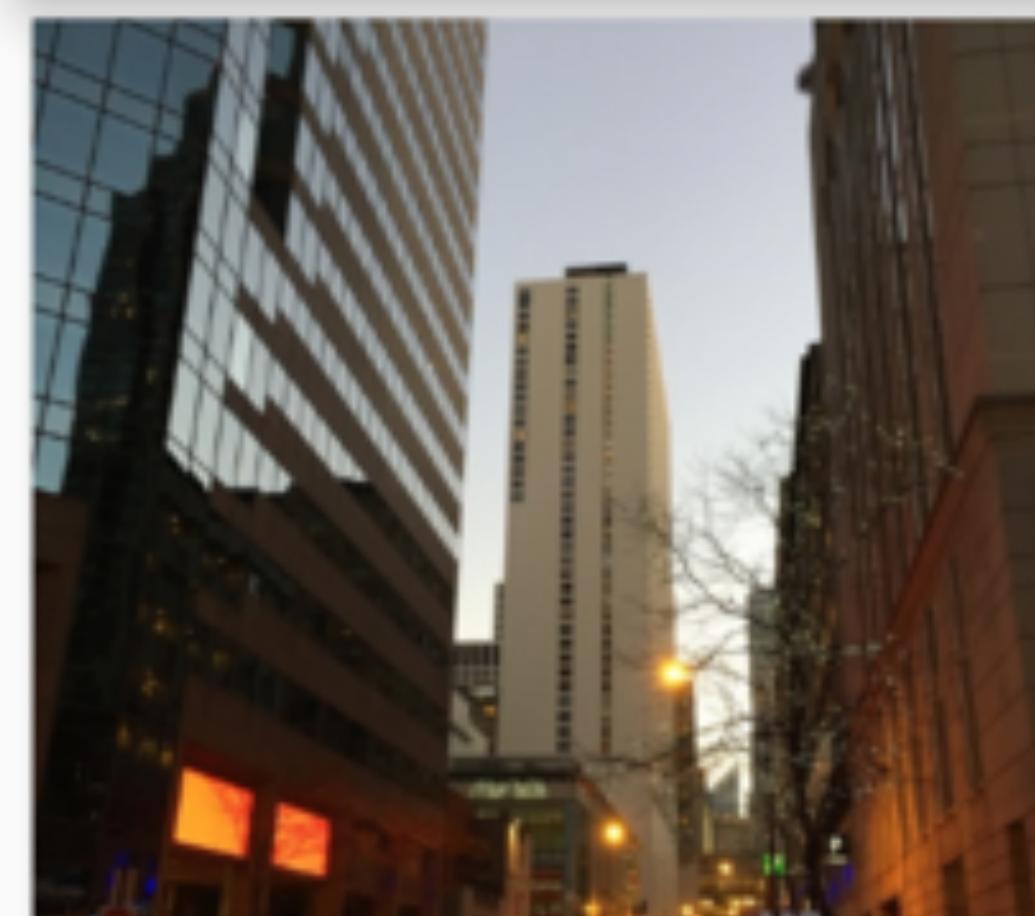
View On Chicago River / Chicago / 2016

21/11/2016 11:11:21  
iPhone SE  
Chicago, Illinois, USA



Four Points Sheraton / Chicago / 2016

20/11/2016 05:41:30  
iPhone SE  
Chicago, Illinois, USA



**PhotoPrism.org**  
**@browseYourLife**

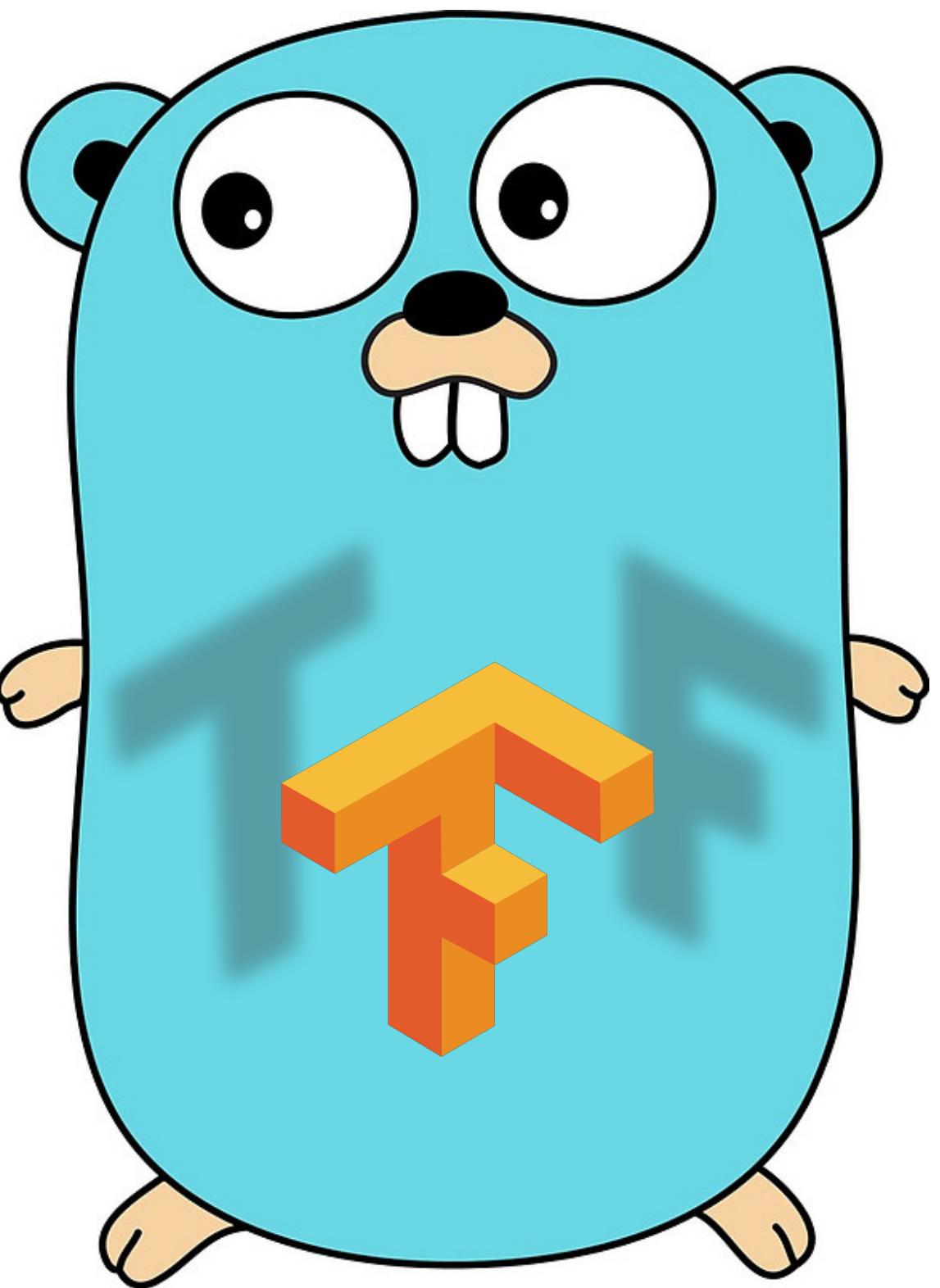
# Get Engaged

Give it a try

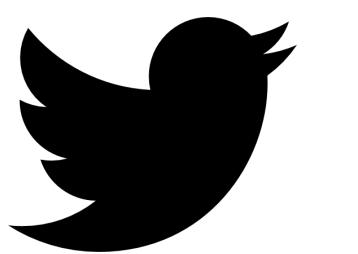
- AutoML: automates ML models design  
[cloud.google.com/automl](https://cloud.google.com/automl)
- TF Hub: repo for modules  
[github.com/tensorflow/models](https://github.com/tensorflow/models)
- A curated list of dedicated resources  
[github.com/jtoy/awesome-tensorflow](https://github.com/jtoy/awesome-tensorflow)

Be part of the community

- [tensorflow.org/community](https://tensorflow.org/community)



Thank  
You!



@NataliePis