iD-Tech

Nicholas Lacroix, Akuang Saechao, Tipper Truong

Overview

- Introduction
- Motivation
- Problem Statement
- Architecture
- Technology
- Demo
- Conclusion

Introduction

Tipper Truong

Nicholas Lacroix

Akuang Saechao

Motivation

- Boom in Machine Learning and AI Topics
- Learn and Understand Data Science Standards
  - Neural Networks and Deep Learning
- Gain Experience in Practical Industry Frameworks
- Create a baseline API for web based machine learning

Problem Statement

Design and develop an application for real world scenarios:

- Self Driving Vehicles
- Real time License plate identification
- Real-time Data Analytics

Problem Statement Part 2

Design and develop an application for real world scenarios:

- Create an application and GUI that can be iterated on to create API's curated for developers' specific needs

Technology

django

python™

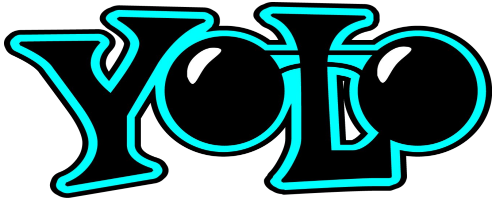
Flask
web development,
one drop at a time

DeepDetect

Darknet
ω
φ
x
Hidden Layers
Hyperparameter

TensorFlow

MySQL®


YOLO

DeepDetect https://deepdetect.com

- An open source deep learning server that enables REST API calls to perform machine learning tasks
- DeepDetect server can be setup with Caffe, TensorFlow, and XGBoost
- Used for image classification, object detection, and text
- Range of built-in model assessment measures (e.g. F1, multiclass log loss, …)
- Support for multiple Machine Learning services, training and prediction calls in parallel
- Supervised learning, regression, and prediction over images and other numerical and textual data, autoencoders, object detection

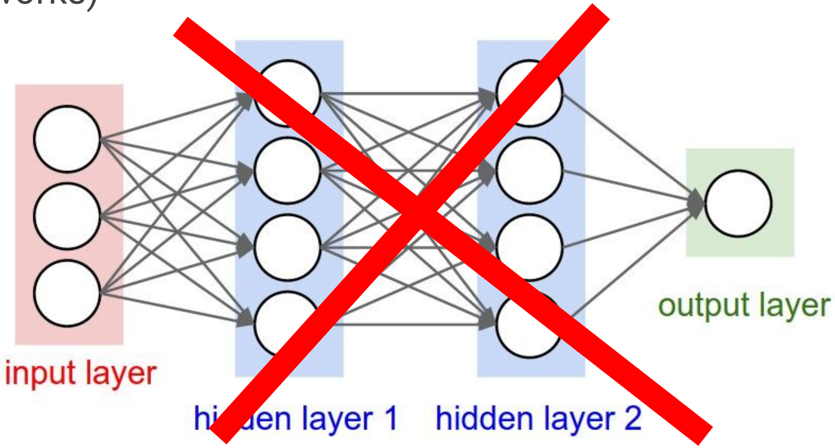
Deepdetect

A combination of Caffe, Tensorflow, and XGBoost is used

- Custom version of Caffe provides the learning via
  - logistic regression, multi-layer perceptron, convolutional networks, recurrent nets, and more.
- XGBoost uses gradient boosted trees to form decision making trees in prediction and learning
  - Fixed sized data sets
  - Disjoints the tree to and compares the values of the data to the weights
  - Line search through the decision tree to minimize the loss function

DarkNet Yolo https://pjreddie.com/darknet/yolo/

- Deep-learning real-time object detection framework
- Faster than CNN and R-CNN (Convolution neural networks)
    - Why?
    - Uses only one neural network
    - Looks at the image once
    - Hence… **Y**ou **O**nly **L**ook **O**nce
- Behind-the-Scenes
    - Image inputted is reduced to **PxP** pixels
    - Divides the image to grids by **SxS**
    - Each grid predicts **B** bounding boxes
    - Calculates and predicts each cell
        - **B * 5 + C**
        - Prediction
            - x, y, width, height for the bounding box rectangle
            - Confidence score
            - Probability distribution over the classes



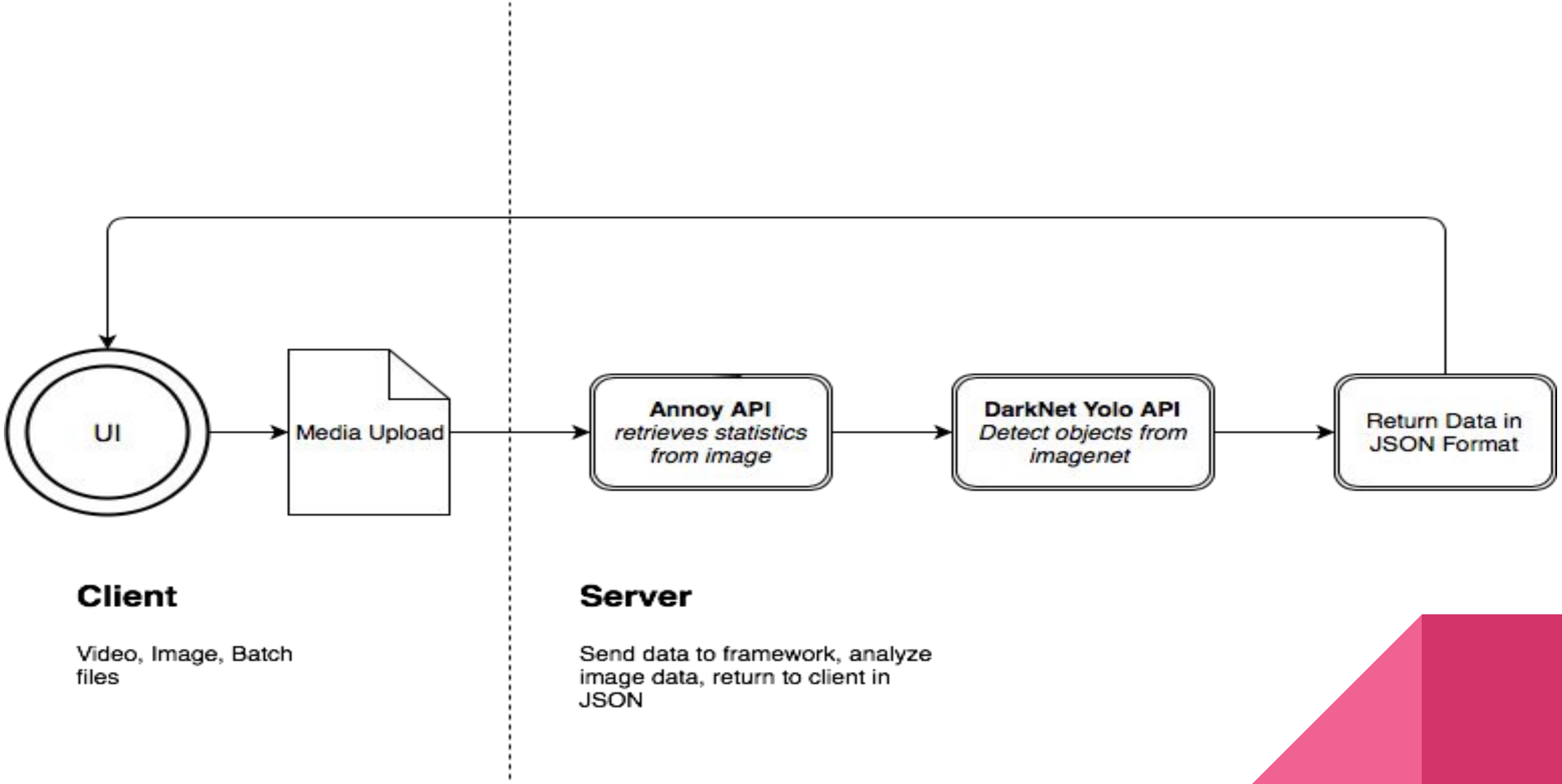
Darkflow https://github.com/thtrieu/darkflow

- Fork of DarkNet with implementation of TensorFlow
- Written in python as a wrapper on top of DarkNet which was written in C in order to utilize TensorFlow
- Highly optimized and lightweight
- Utilizes the neural network designed in DarkNet
- Combines the TensorFlow with the convolutional layers and weights used in DarkNet Yolo
- Provides additional support for real time mobile predictions through openCV
    - Difference: Darknet does not have openCV packaged into it
        - Darkflow reduces overhead compared to Darknet

Annoy

- Main large data statistic manager created by spotify
  - Allows for k-means distributions and displayable to the screen
  - Uses description vectors from images processed via darknet and/or darkflow
    - These are used to find nearest neighbors, thus finding the closest looking image from a image set in comparison to the uploaded image
  - Uses Euclidean distance for extra optimization

Architecture


UI
Media Upload
Annoy API
retrieves statistics from image
DarkNet Yolo API
Detect objects from imagenet
Return Data in JSON Format

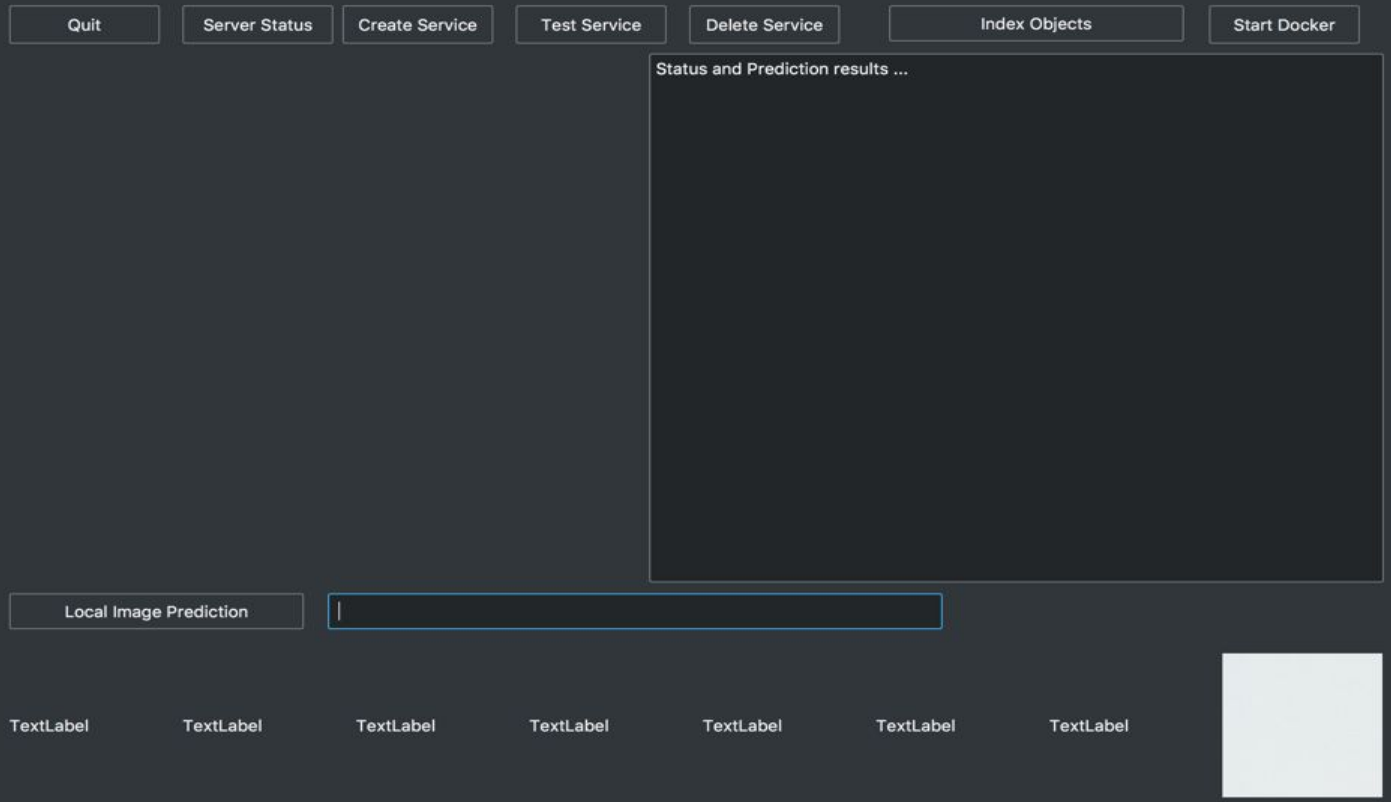
Client

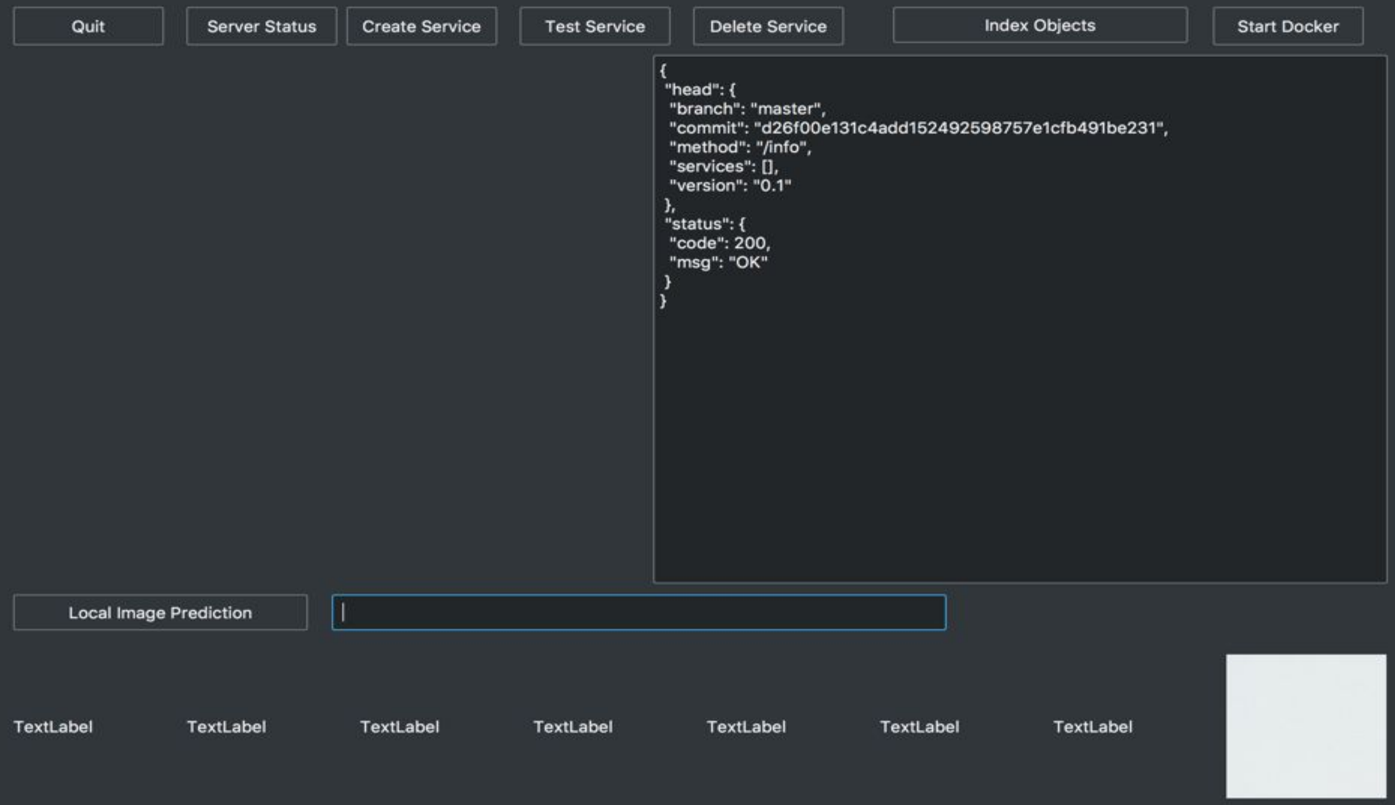
Video, Image, Batch files

Server

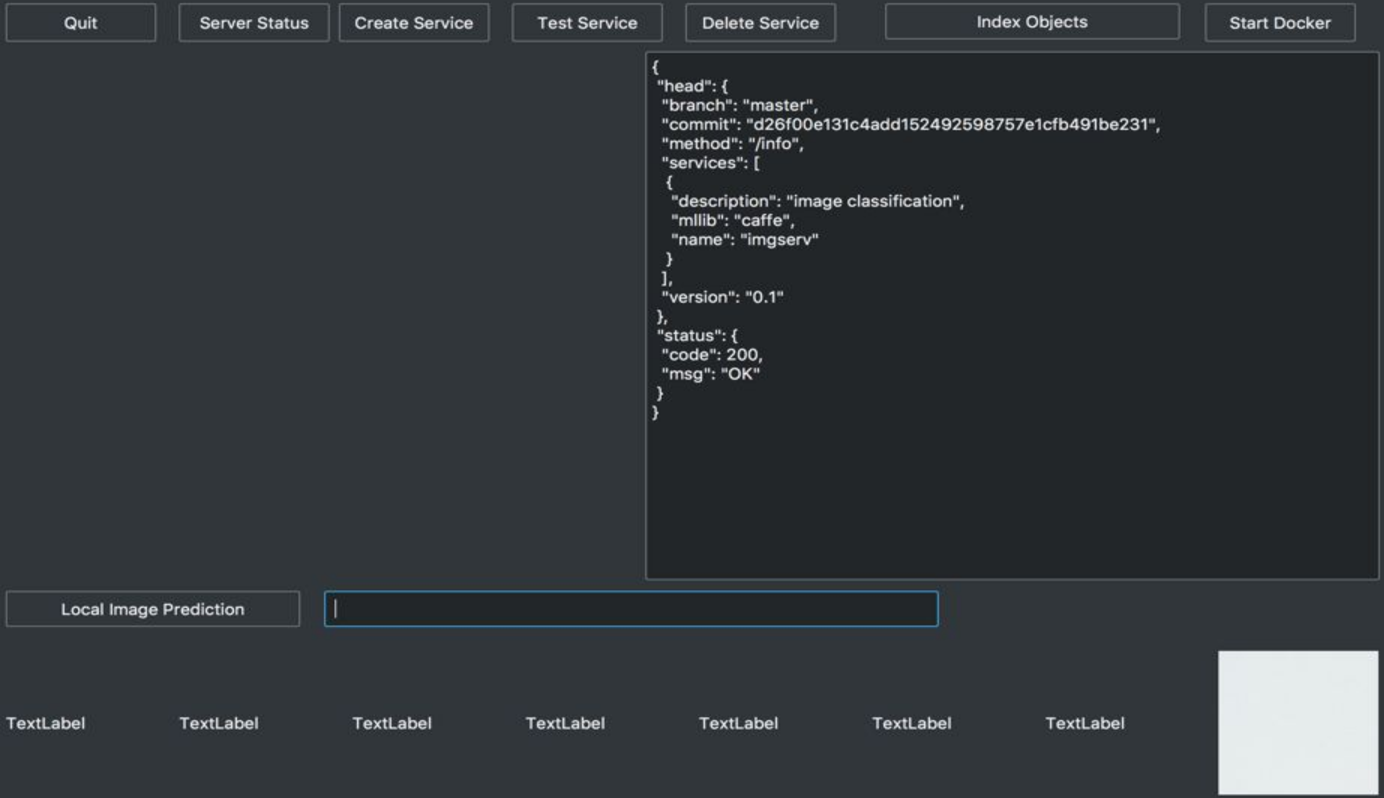
Send data to framework, analyze image data, return to client in JSON

Start
DEMO

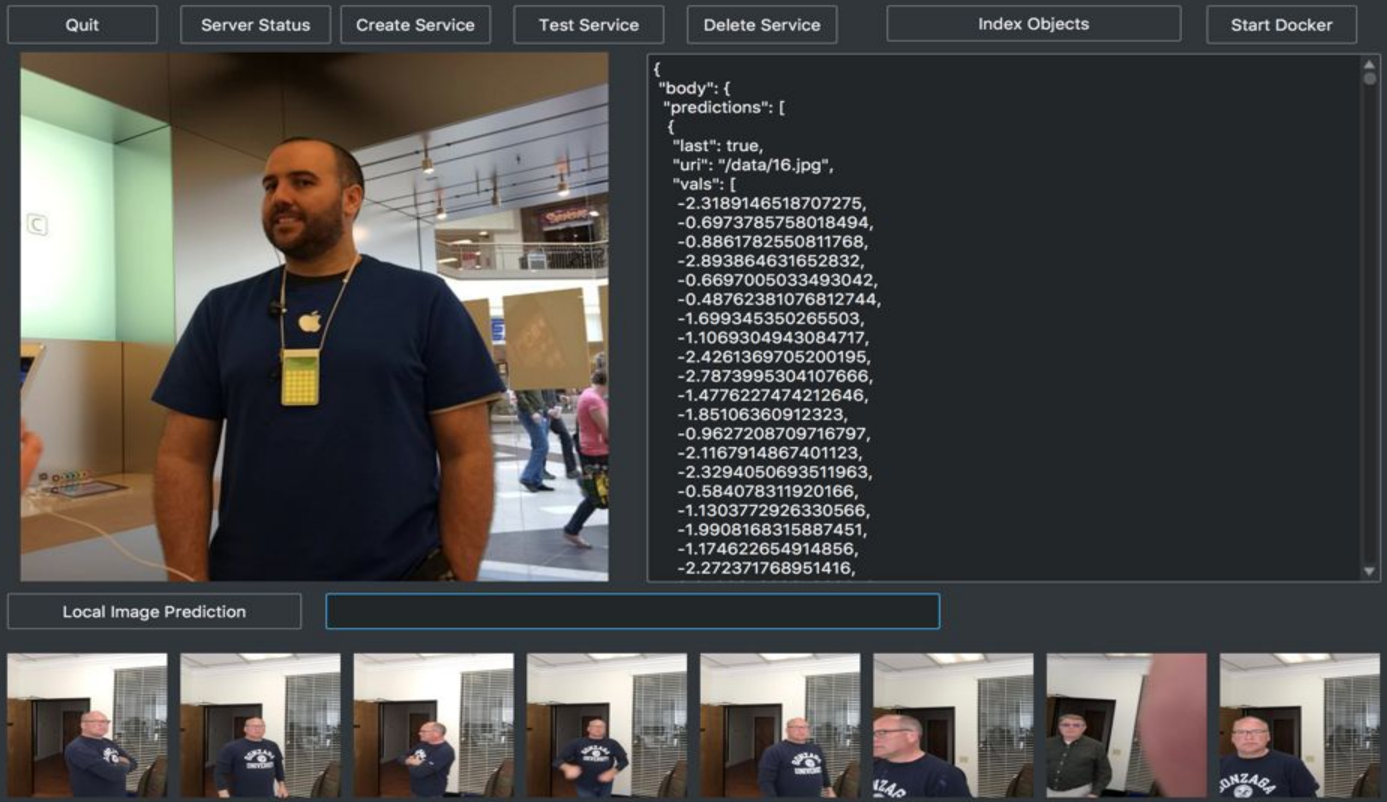

Quit
Server Status
Create Service
Test Service
Delete Service
Index Objects
Start Docker
Status and Prediction results ...
Local Image Prediction
TextLabel
TextLabel
TextLabel
TextLabel
TextLabel
TextLabel
TextLabel



```
{
 "head": {
  "branch": "master",
  "commit": "d26f00e131c4add152492598757e1cfb491be231",
  "method": "/info",
  "services": [],
  "version": "0.1"
 },
 "status": {
  "code": 200,
  "msg": "OK"
 }
}
```



```
{
 "head": {
  "branch": "master",
  "commit": "d26f00e131c4add152492598757e1cfb491be231",
  "method": "/info",
  "services": [
   {
    "description": "image classification",
    "mllib": "caffe",
    "name": "imgserv"
   }
  ],
  "version": "0.1"
 },
 "status": {
  "code": 200,
  "msg": "OK"
 }
}
```



```
{
 "body": {
  "predictions": [
   {
    "last": true,
    "uri": "/data/16.jpg",
    "vals": [
     -2.3189146518707275,
     -0.6973785758018494,
     -0.8861782550811768,
     -2.893864631652832,
     -0.6697005033493042,
     -0.48762381076812744,
     -1.699345350265503,
     -1.1069304943084717,
     -2.4261369705200195,
     -2.7873995304107666,
     -1.4776227474212646,
     -1.85106360912323,
     -0.9627208709716797,
     -2.1167914867401123,
     -2.3294050693511963,
     -0.584078311920166,
     -1.1303772926330566,
     -1.9908168315887451,
     -1.174622654914856,
     -2.272371768951416,
```

Future Implementation

- Deploy to Amazon EC2
- Train new models

QUESTIONS?