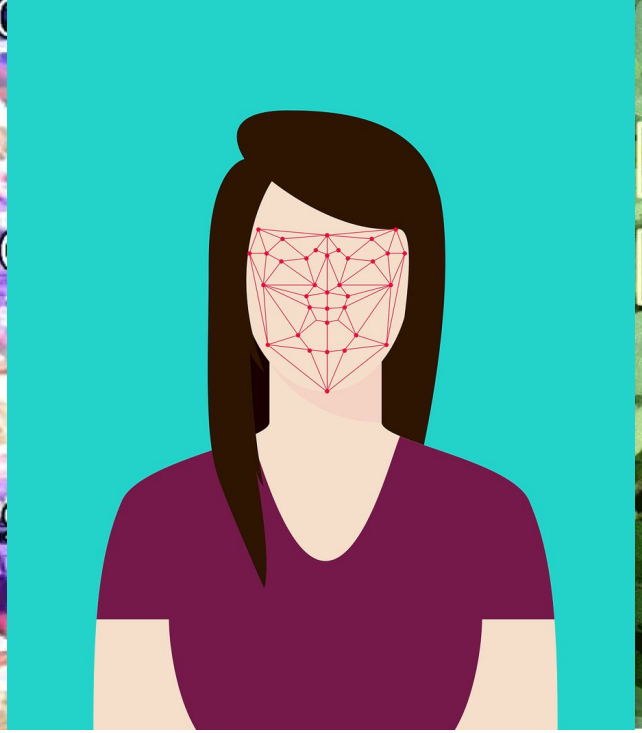



Deep Learning in Financial Inclusion

Data Science Nigeria 2018 Bootcamp | Matt Grasser, BFA | 10 Oct 2018

Who are BFA/Matt?

BFA is a global consulting firm specialized in using finance to create solutions for low-income people by integrating:

Customer Insights

HCD, UI/UX research, usability testing, ethnography, surveys, focus groups and financial diaries

Inclusive Fintech

Application of APIs, machine learning, bitcoin/blockchain, lean startup, experiments, MVPs and prototyping

Policy & Ecosystem Development

RegTech, digital identity, data privacy, regulatory impact assessment, design of policy and regulatory frameworks to enable innovation, ecosystem development

Business Insights

Quantitative analytics, market segmentation, revenue potential, pricing and channel development, costing and efficiency

Finance for Life

Financial solutions in health, education, sanitation, housing and productivity in agriculture and small business

BFA

Let’s Talk about Deep Learning!

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

Conclusion
Recap, Q&A, Thoughts

Topics we'll aim to cover in today's bootcamp session...

BFA

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

Conclusion
Recap, Q&A, Thoughts

Topics we'll aim to cover in today's bootcamp session...

BFA

What is Deep Learning?

Dictionary Definition

Deep learning is a class of machine learning algorithms that:

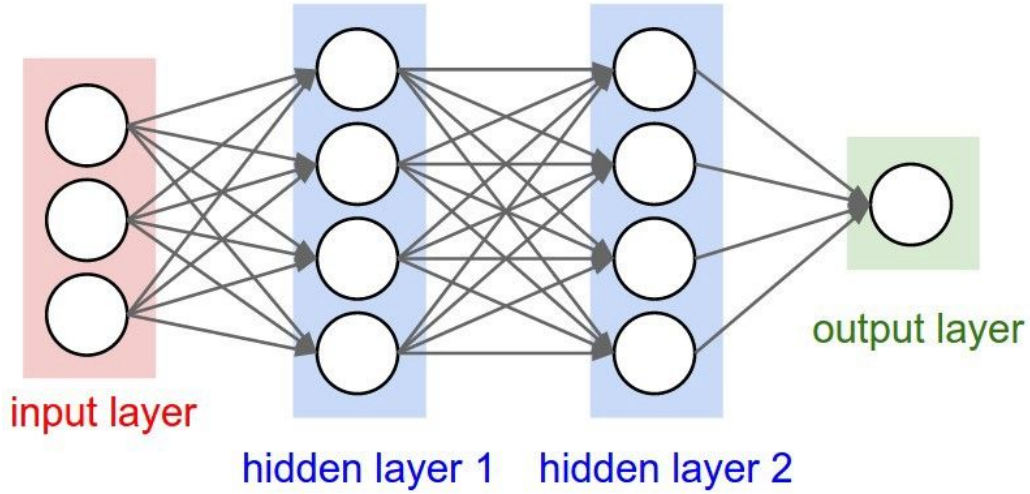
- Use a cascade of **multiple layers** of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- **Learn** in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a **hierarchy** of concepts.

What is Deep Learning?

Dictionary Definition

Deep learning is a class of machine learning algorithms that:

- Use a cascade of **multiple layers** of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.

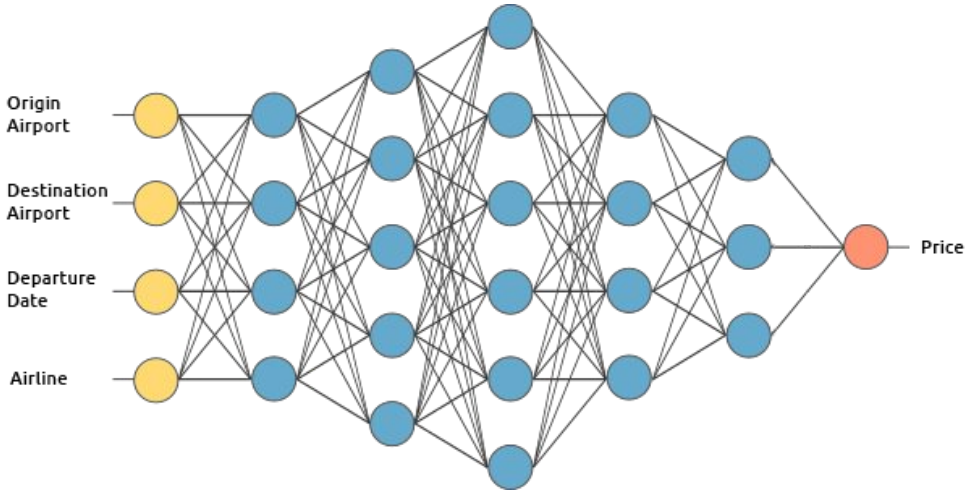


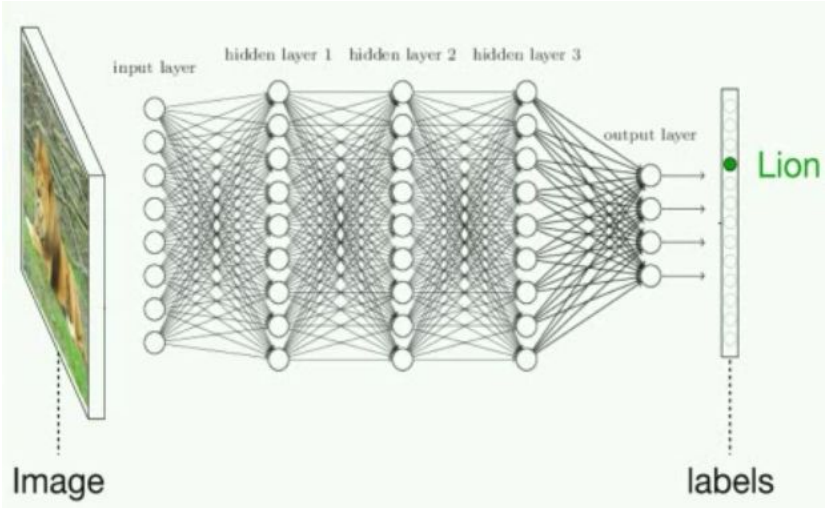
What is Deep Learning?

Dictionary Definition

Deep learning is a class of machine learning algorithms that:

- Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a **hierarchy** of concepts.



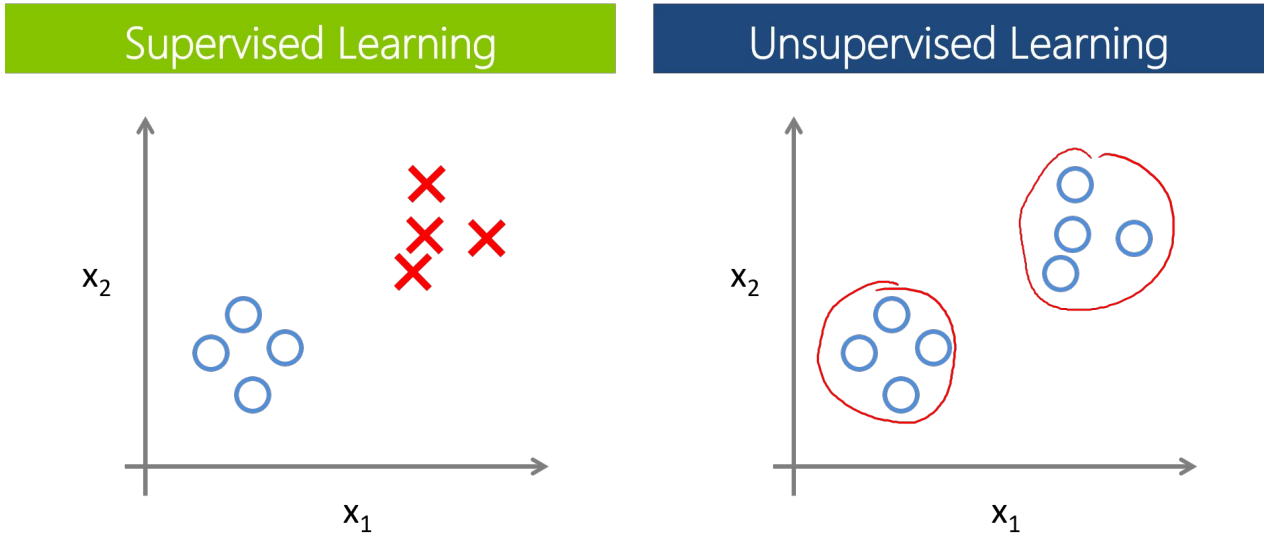


What is Deep Learning?

Dictionary Definition

Deep learning is a class of machine learning algorithms that:

- **Learn** in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.



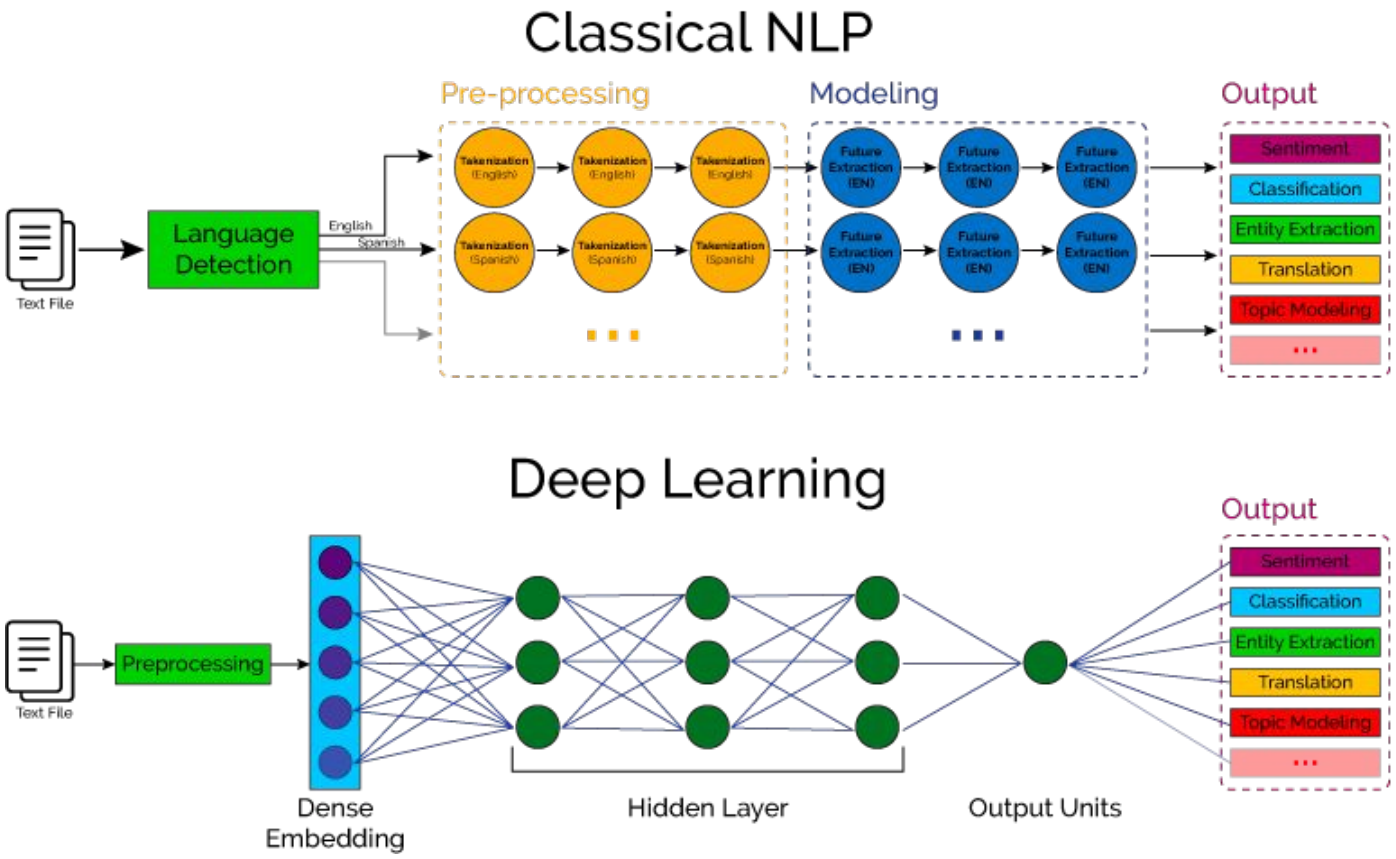
Applications of DL?

Sample of Useful Applications

Deep learning is used for:

- Automatic Speech/Sound Recognition
- Image Recognition
- Natural Language Processing
- Recommendation Systems
- Financial Fraud Detection
- Bioinformatics
- ...and much more!

Natural Language Processing: Example of Before & After DL



Applications of DL?

Sample of Useful Applications

Applications of DL?

Sample of Useful Applications

Deep learning is used for:

- Automatic Speech/Sound Recognition
- **Image Recognition and Computer Vision**
- Natural Language Processing
- Recommendation Systems
- Financial Fraud Detection
- Bioinformatics
- ...and much more!

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

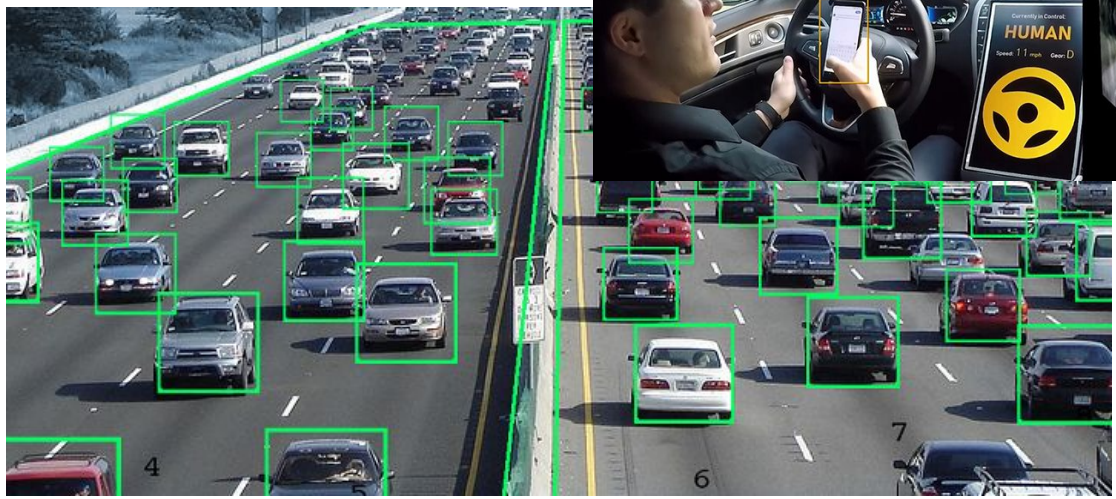
Conclusion
Recap, Q&A, Thoughts

Topics we'll aim to cover in today's bootcamp session...

BFA

Computer Vision (CV)

- Definition
- Types of CV
- Key Concepts
- Architectures



BFA

Computer Vision (CV)

- **Definition**
- Types of CV
- Key Concepts
- Architectures

Computer Vision is:

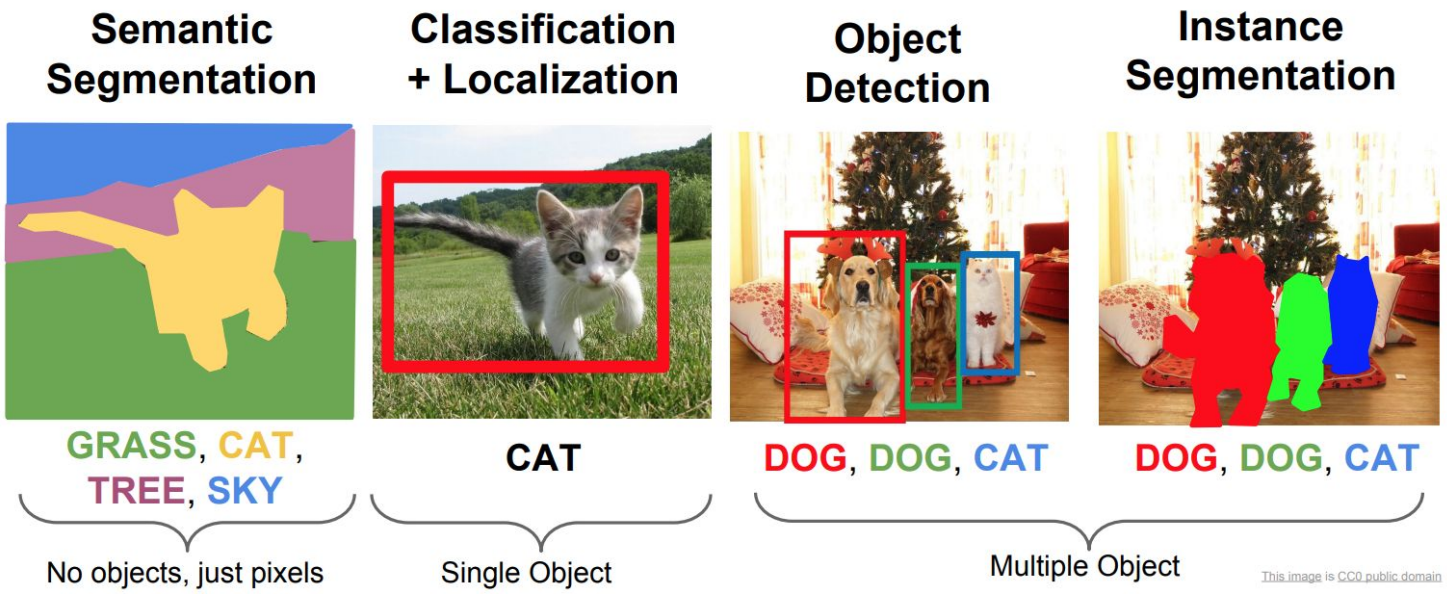
An **interdisciplinary** field that deals with how computers can be made to gain high-level **understanding from digital images or videos**. From the perspective of engineering, it seeks to augment, and ultimately automate tasks that the human visual system can do.

Computer Vision (CV)

- Definition
- **Types of CV**
- Key Concepts
- Architectures

Types of Computer Vision:

- Semantic Segmentation: no objects, just pixels
- Classification + Localization: single object
- Object Detection: multiple objects, boxed
- Instance Segmentation: multiple objects, masking



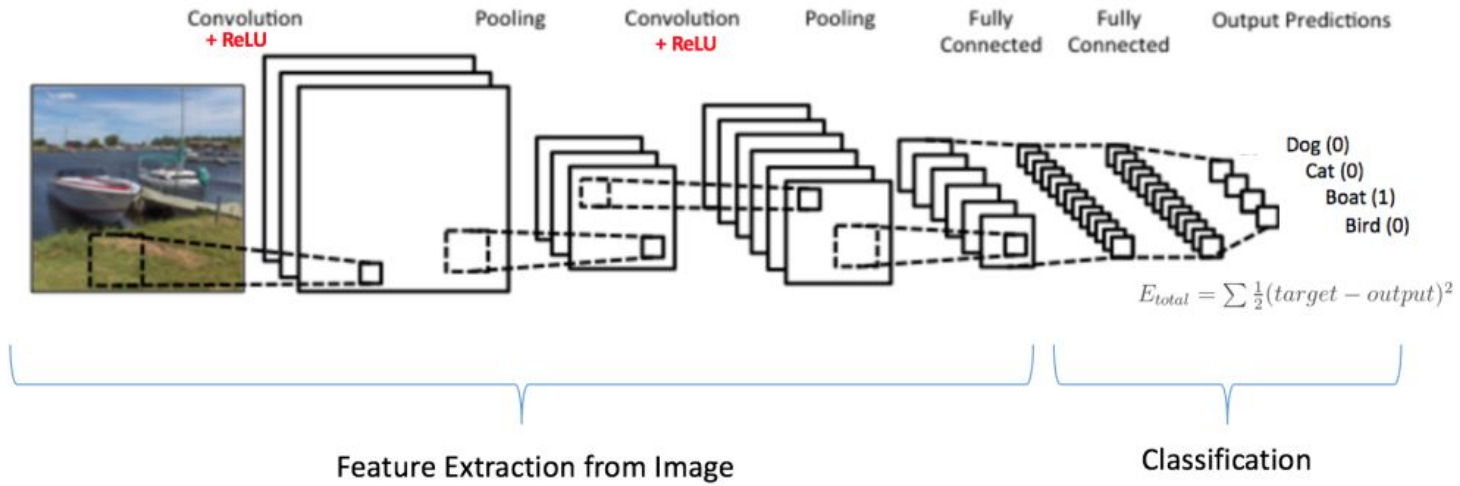
BFA

Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

Four Simple Building Blocks:

- Convolutions (CONV)
- Rectified Linear Units (RELU)
- Pooling (POOL)
- Fully Connected (FC)



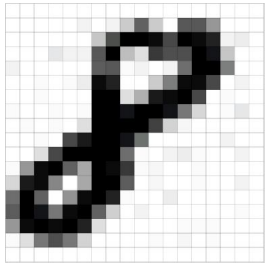
BFA

Computer Vision (CV)

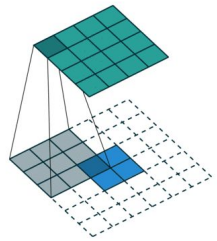
- Definition
- Types of CV
- **Key Concepts**
- Architectures

1. Convolutions:

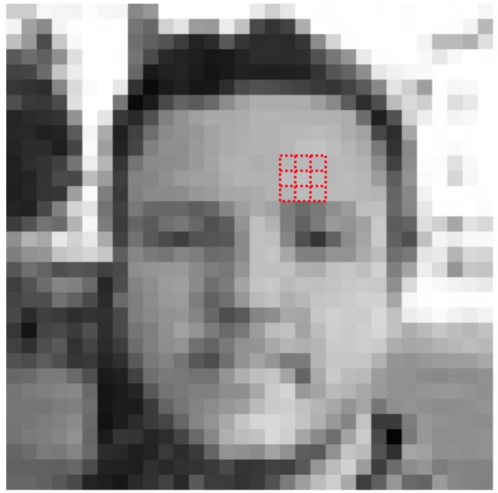
- Applying a **linear operation across several layers of ordered data** to condense into a new set of layers (i.e. running a “filter” as a “sliding window” across a “volume” to create a deeper volume)



$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$



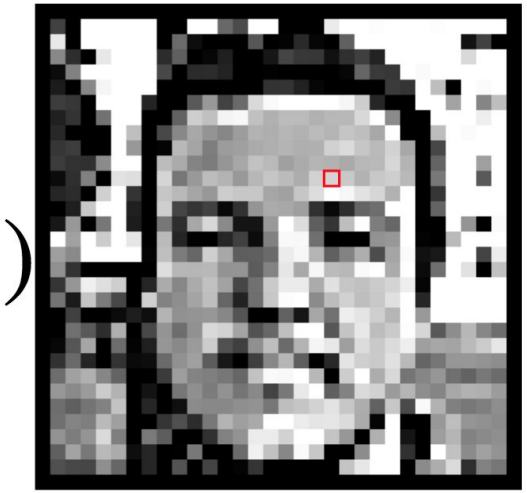
Below, for each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right. Hover over a pixel on either image to see how its value is computed.



input image

$$\begin{pmatrix} 183 \times 0 & + 186 \times -1 & + 185 \times 0 \\ + 189 \times -1 & + 193 \times 5 & + 190 \times -1 \\ + 180 \times 0 & + 187 \times -1 & + 186 \times 0 \end{pmatrix} = 213$$

kernel: sharpen



output image

BFA

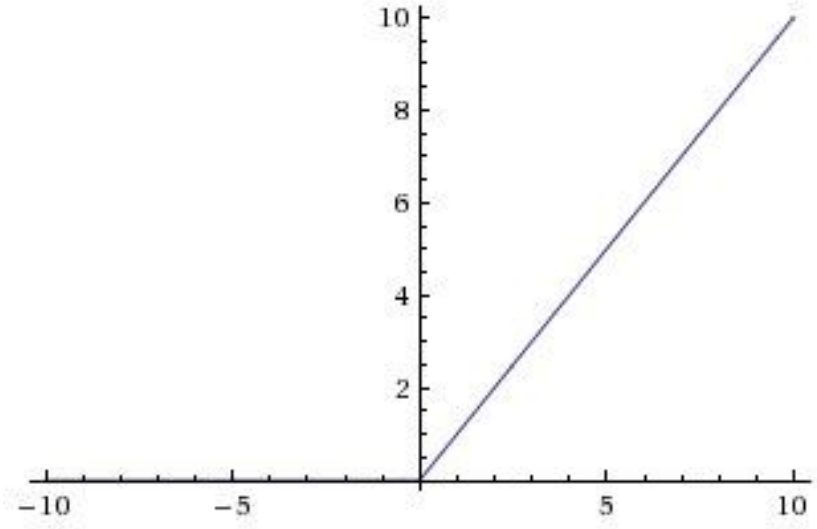
Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

2. Rectified Linear Units:

- After each conv layer, it is convention to apply a **nonlinear "activation" layer**. The purpose of this layer is to introduce via nonlinearity the ability for a model to account for **interaction effects** and **nonlinear effects** that the linear convolutions cannot.
- tl;dr stops the model from collapsing to a single convolution layer

$f(x) = \max(0, x)$



BFA

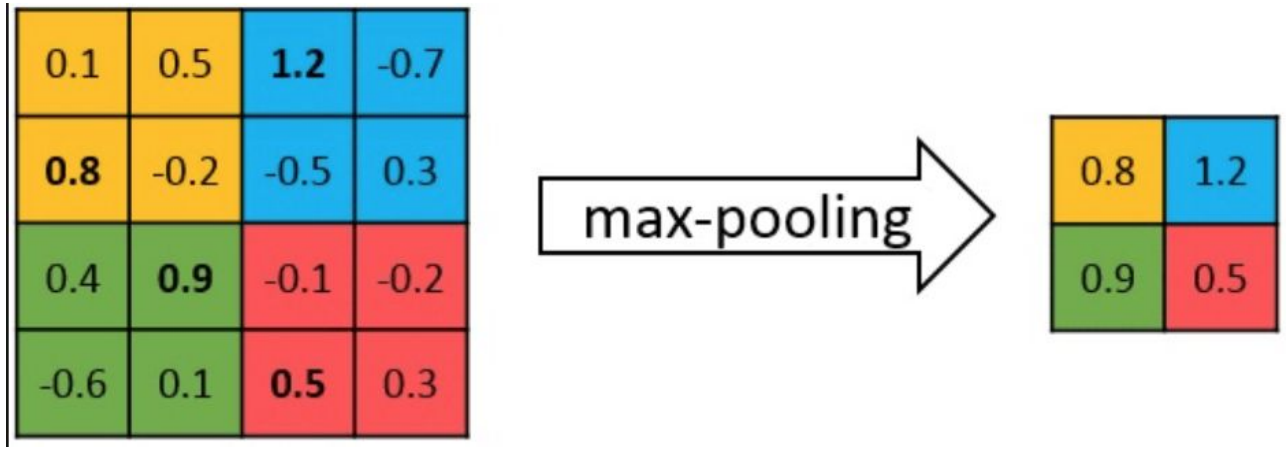
Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

3. Pooling Layers:

The pooling layer is used to **reduce the spatial dimensions**, but not depth, on a convolution neural network, model, basically this is what you gain:

- By having less spatial information you gain computation **performance**
- Less spatial information also means less parameters, so **less chance to over-fit**
- You get some **translation invariance**



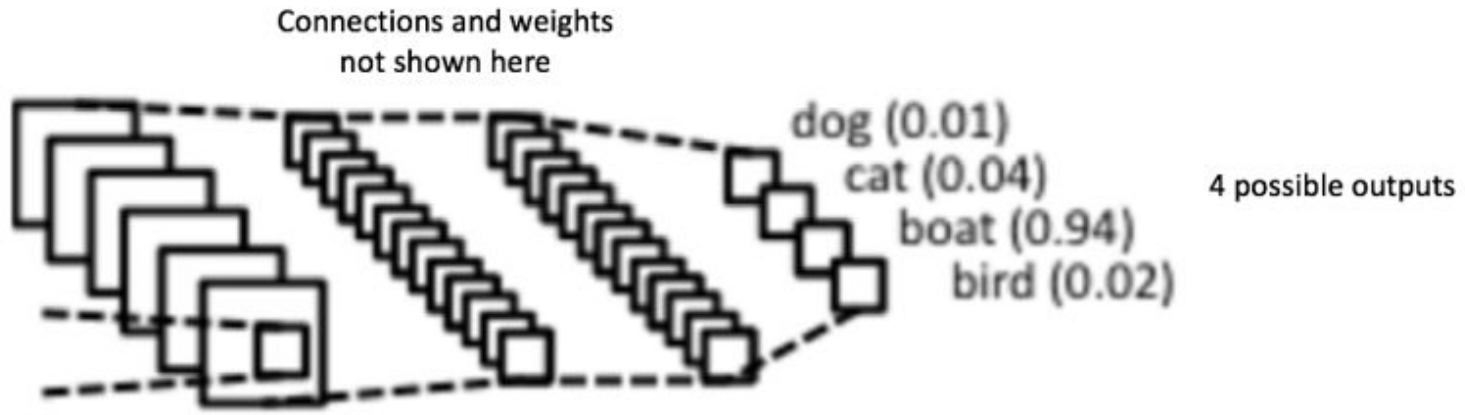
BFA

Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

4. Fully Connected Layers:

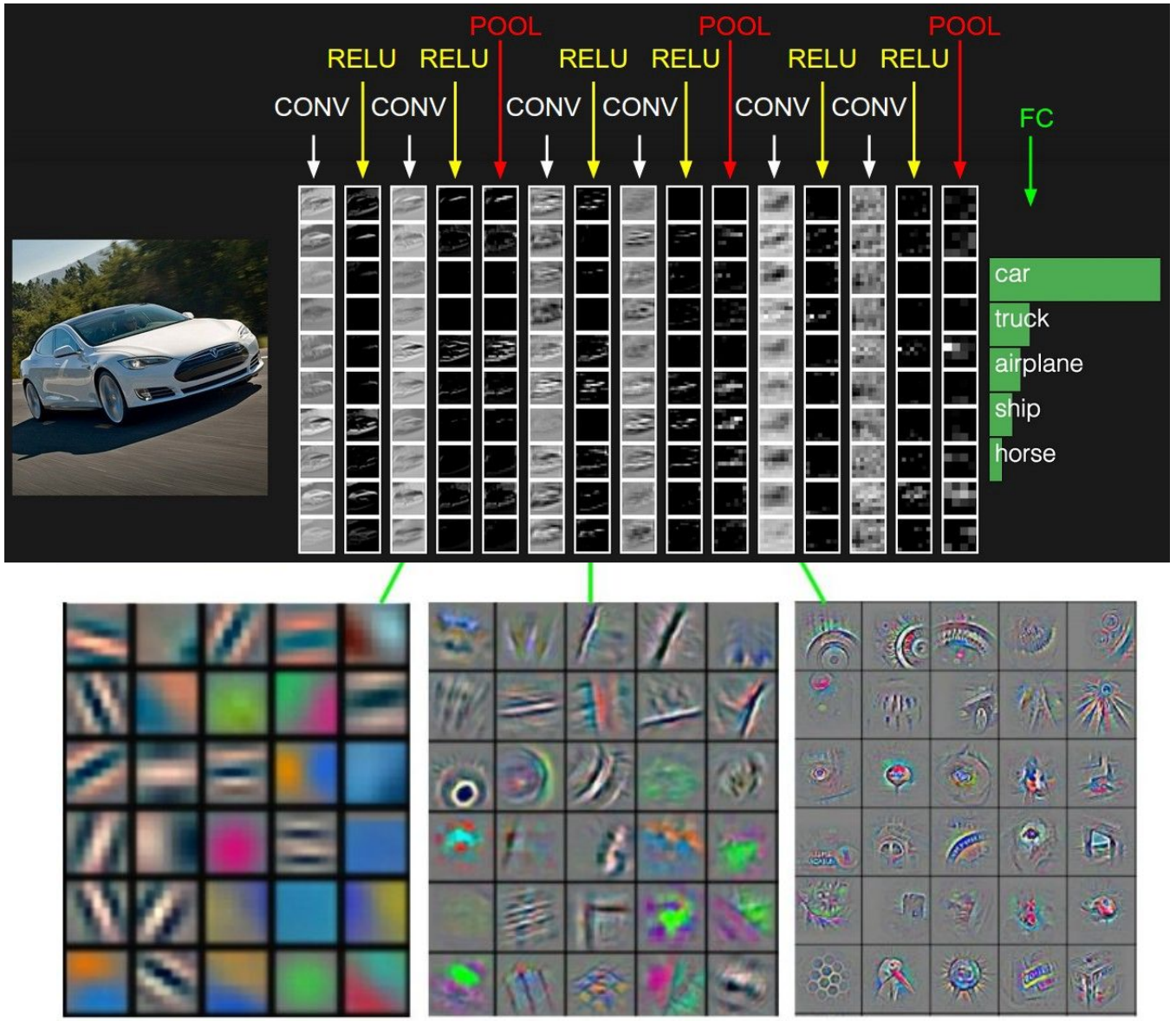
The output from the convolutional and pooling layers represent high-level *features* of the *input* image. The purpose of the Fully Connected layer is to **combine these features** (e.g. into probabilities or one-hot encodings) into **predictions for various classes** based on the training dataset.



Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

Altogether, it looks something like:

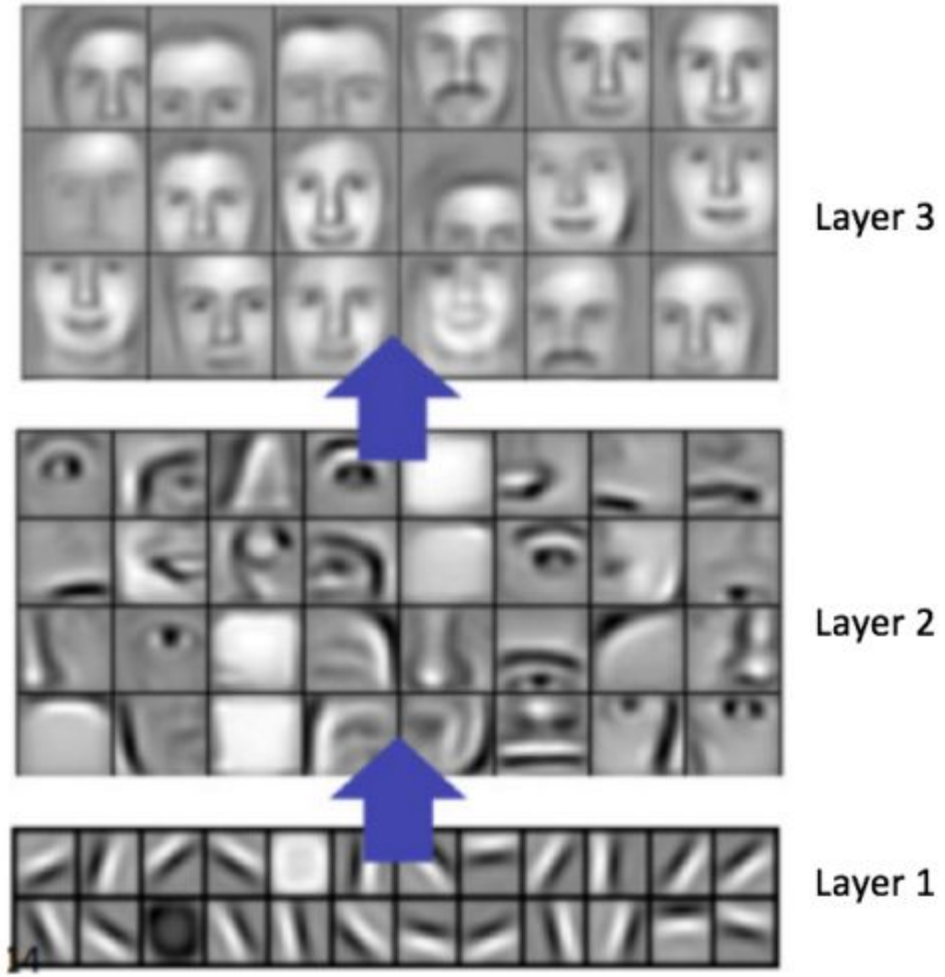


BFA

Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

Or like:

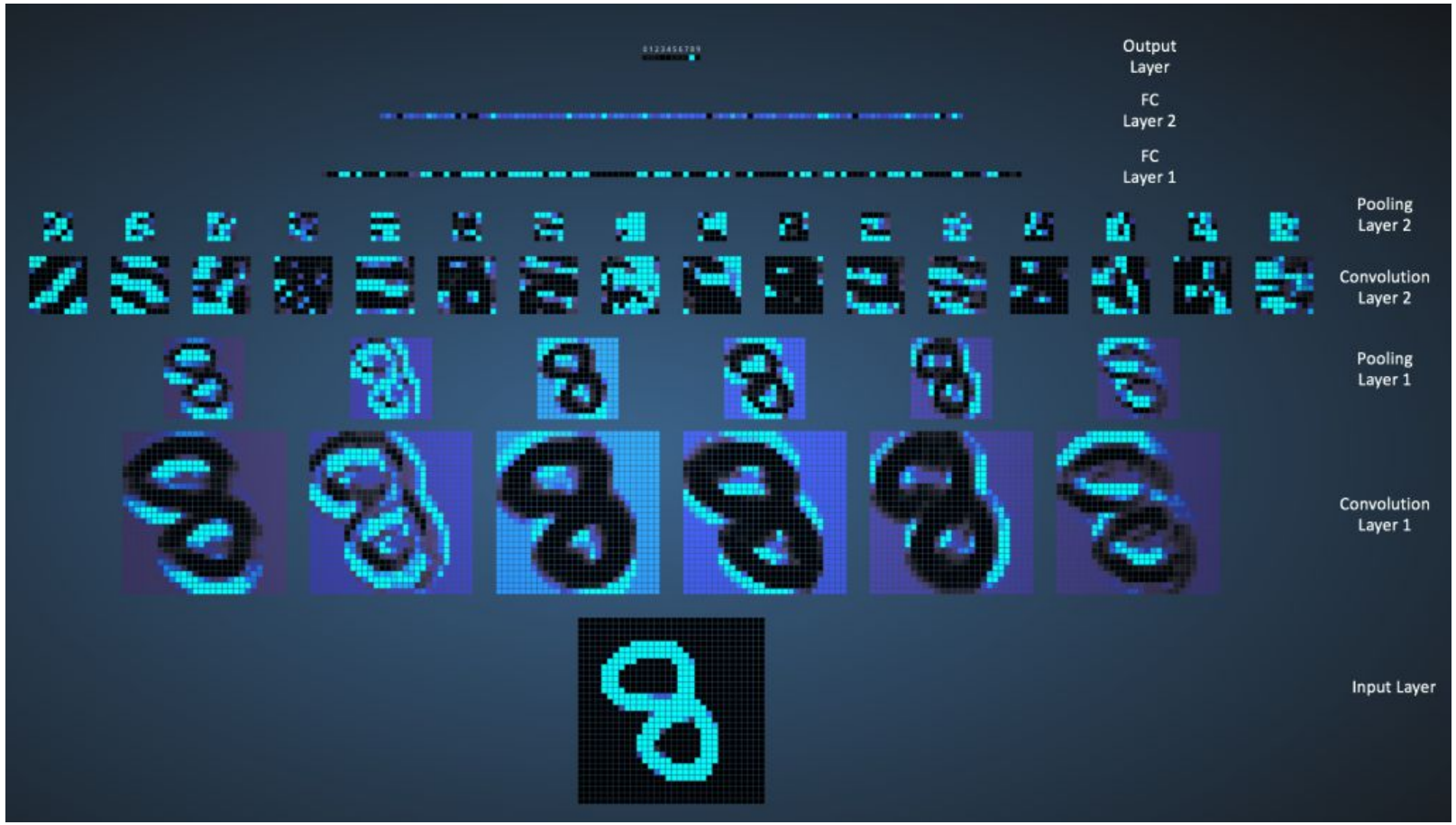


BFA

Computer Vision (CV)

- Definition
- Types of CV
- **Key Concepts**
- Architectures

Or like:



BFA

Computer Vision (CV)

- Definition
- Types of CV
- Key Concepts
- **Architectures**

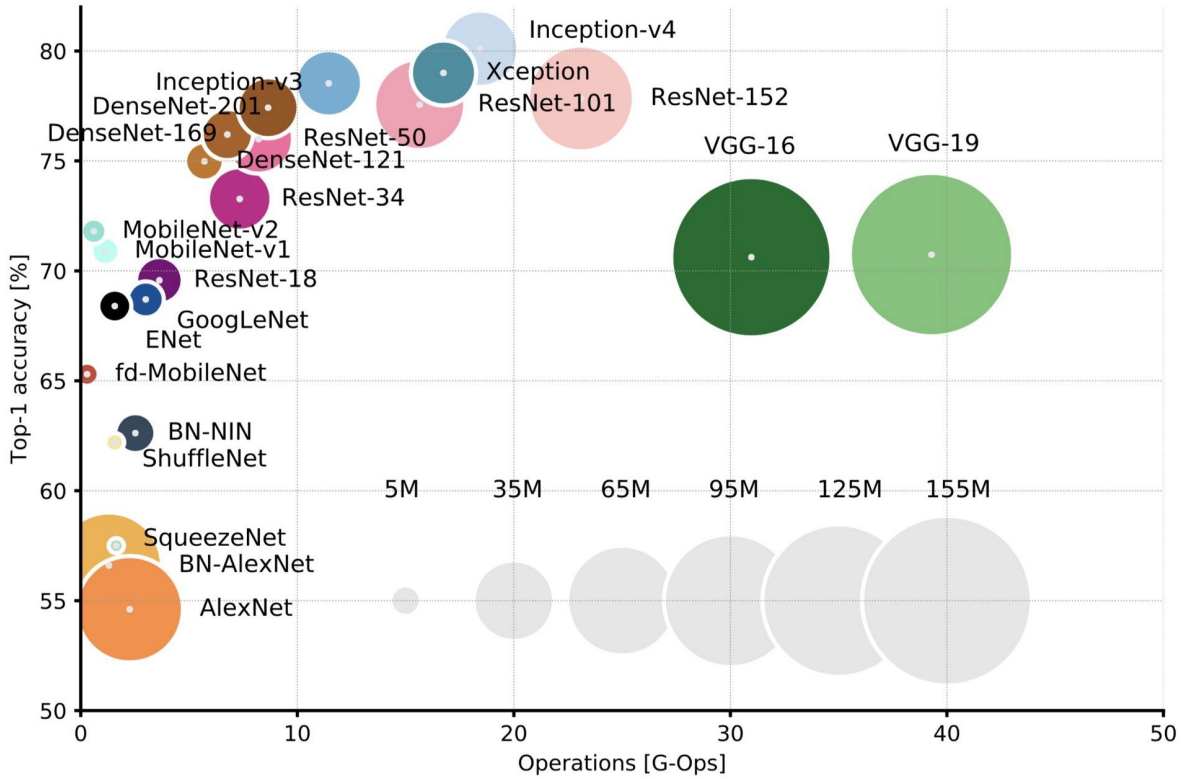
History of ConvNet Architectures:

- **LeNet (1990s)**
- [Incubation Period] (1990s to 2012)
- AlexNet (2012)
- ZF Net (2013)
- GoogLeNet (2014)
- **VGGNet (2014)**
- **ResNets (2015)**
- DenseNet (August 2016)
- **YOLO**, Shufflenet, **Mobilenet**, Xception, Squeezenet (2016-present)

Computer Vision (CV)

- Definition
- Types of CV
- Key Concepts
- **Architectures**

Performance of ConvNet Architectures:

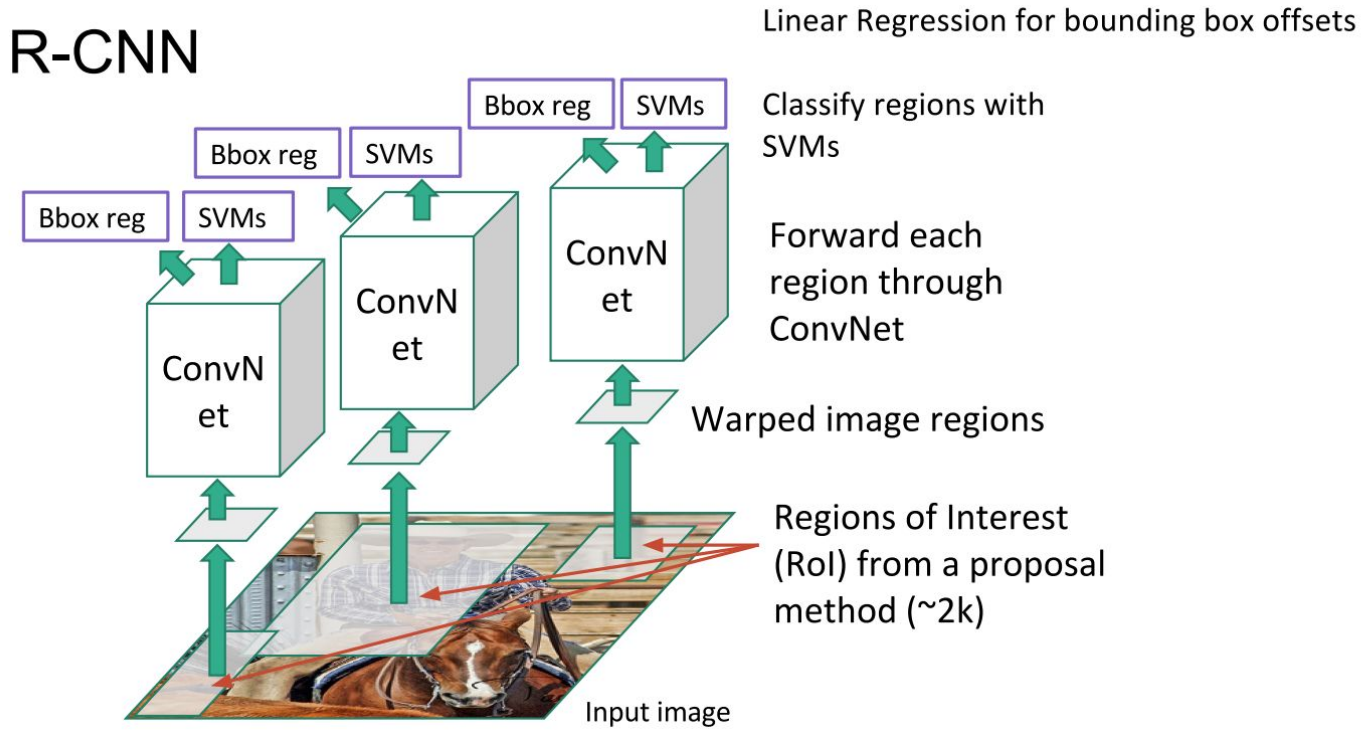


BFA

Computer Vision (CV)

- Definition
- Types of CV
- Key Concepts
- **Architectures**

Relevant to Our Discussion:



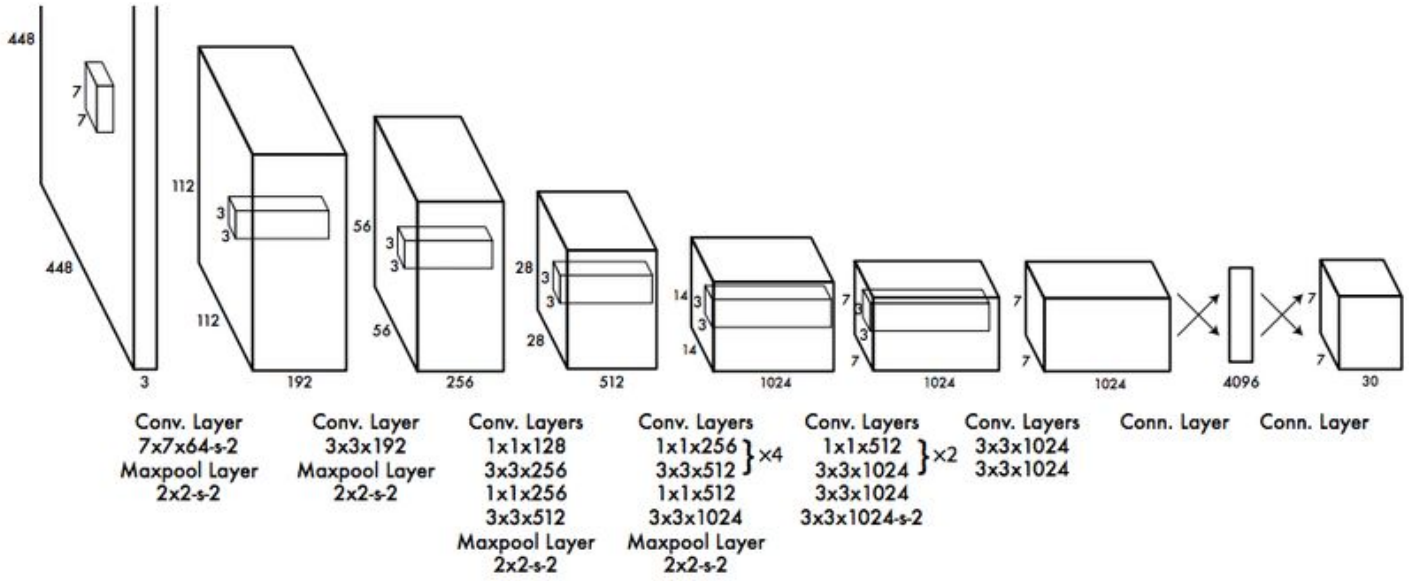
BFA

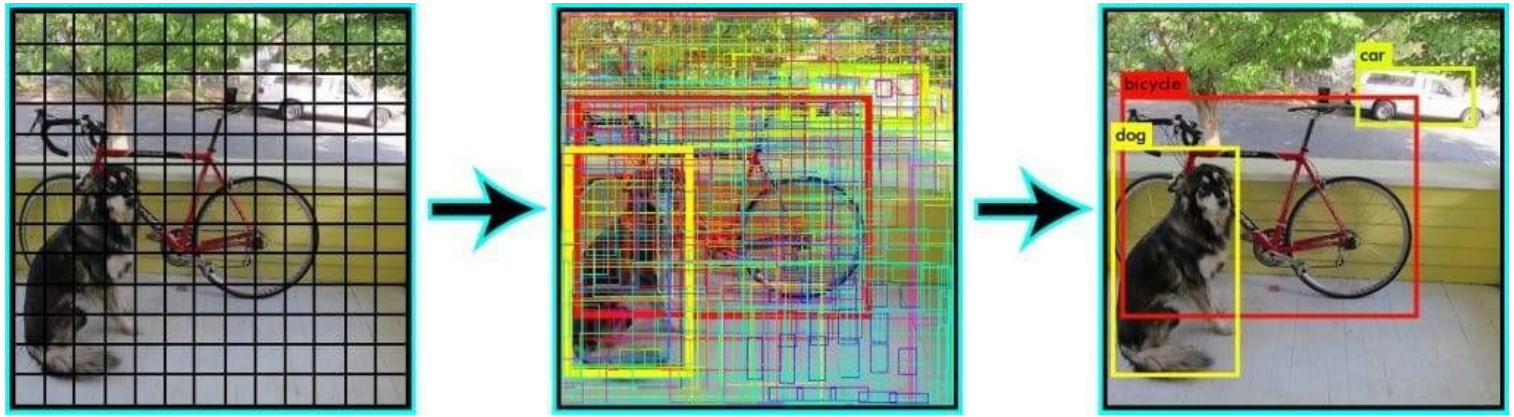
Computer Vision (CV)

- Definition
- Types of CV
- Key Concepts
- **Architectures**

Relevant to Our Discussion:

YOLO





Phew!
Enough Theory?

Let’s move into the fun part: Practical Examples!

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

Conclusion
Recap, Q&A, Thoughts

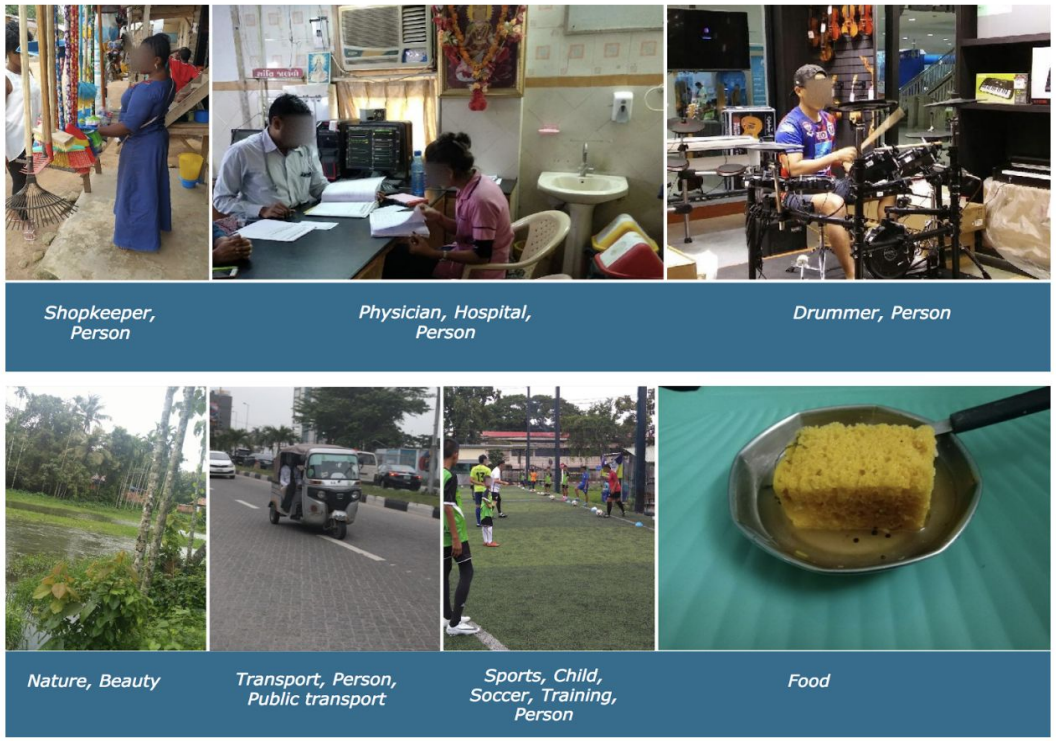
Topics we'll aim to cover in today's bootcamp session...

BFA

Relevant Examples

- **Competitive**
- Academic
- Catalyst Fund
- FIBR

Google AI Inclusive Images (Classification)

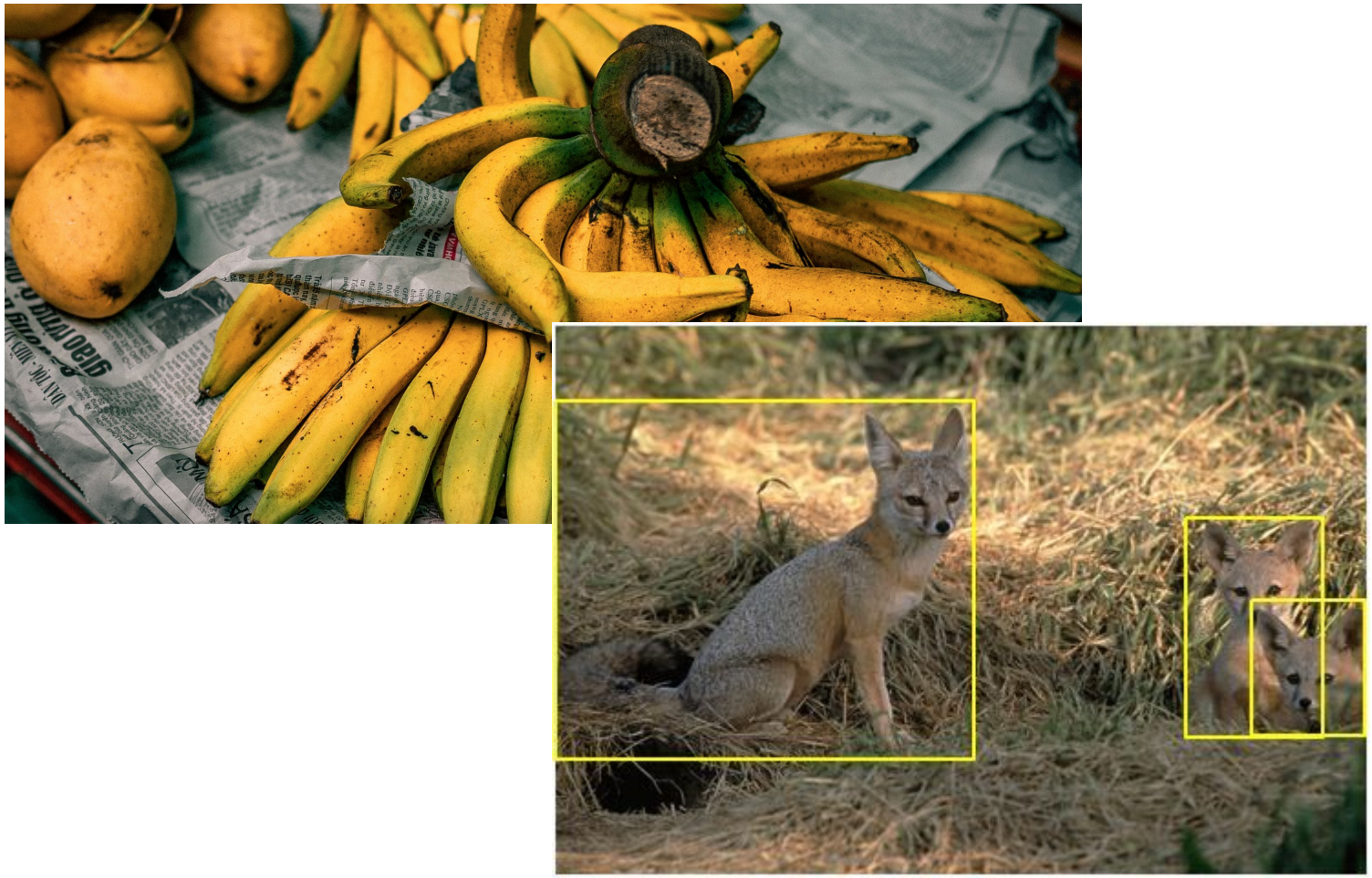


BFA

Relevant Examples

- **Competitive**
- Academic
- Catalyst Fund
- FIBR

ImageNet (Object Localization)

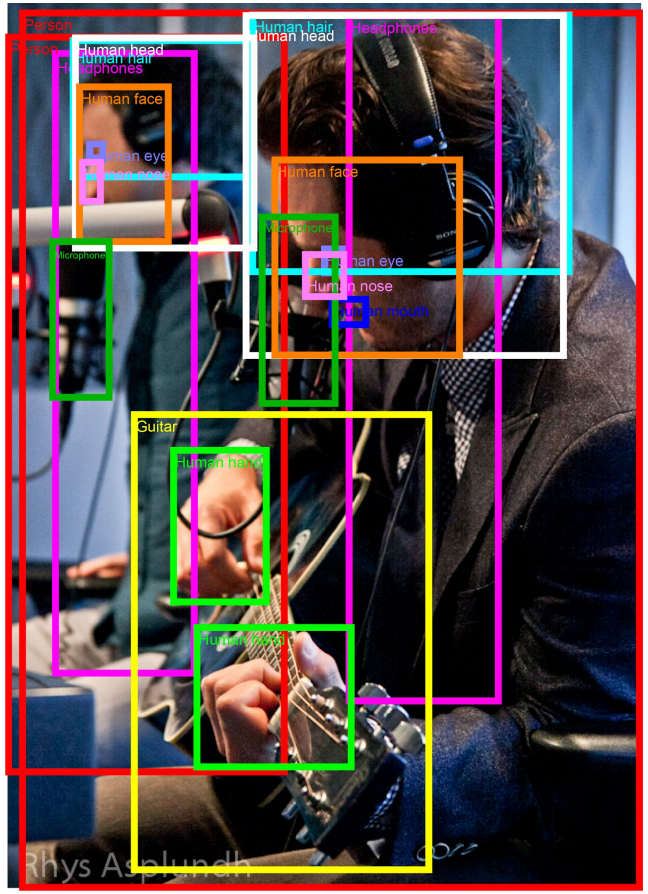


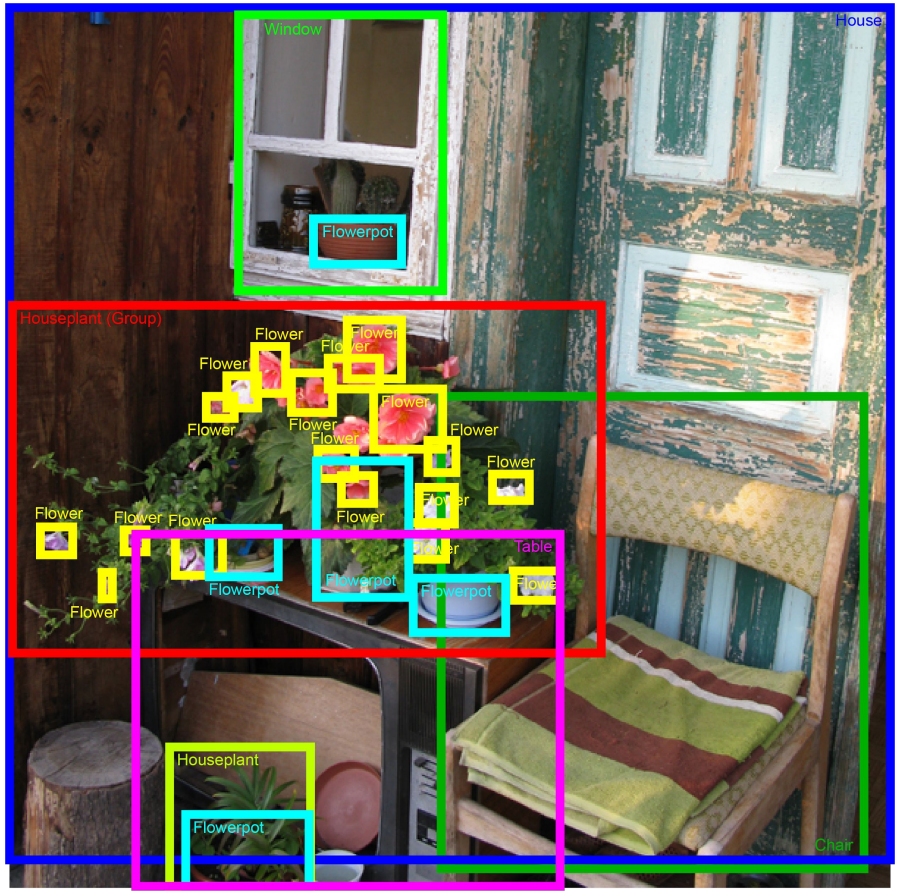
BFA

Google AI Open Images (Object Detection)

Relevant Examples

- **Competitive**
- Academic
- Catalyst Fund
- FIBR



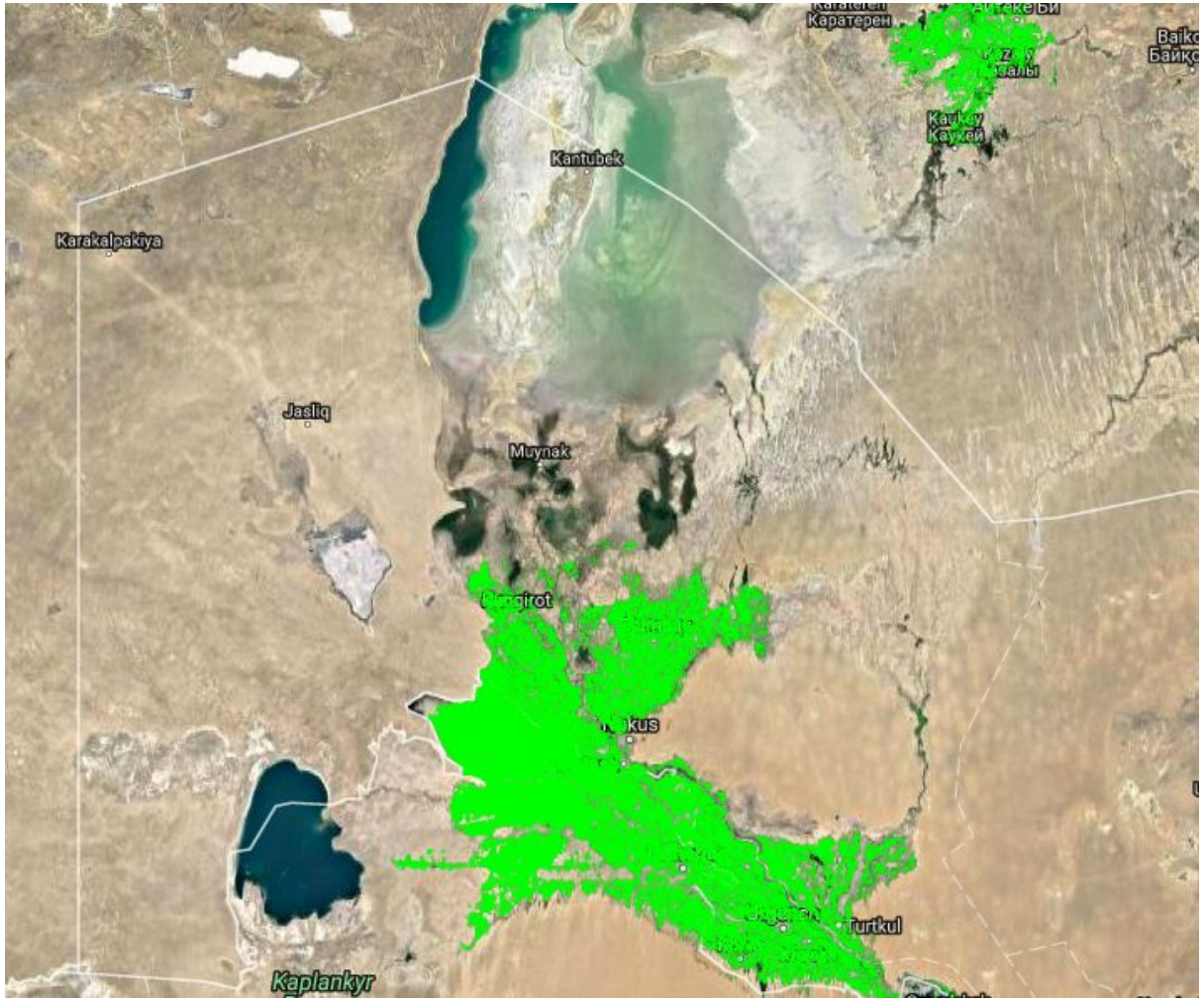


BFA

Segmenting Cropland: UW-Madison + NASA

Relevant Examples

- Competitive
- **Academic**
- Catalyst Fund
- FIBR

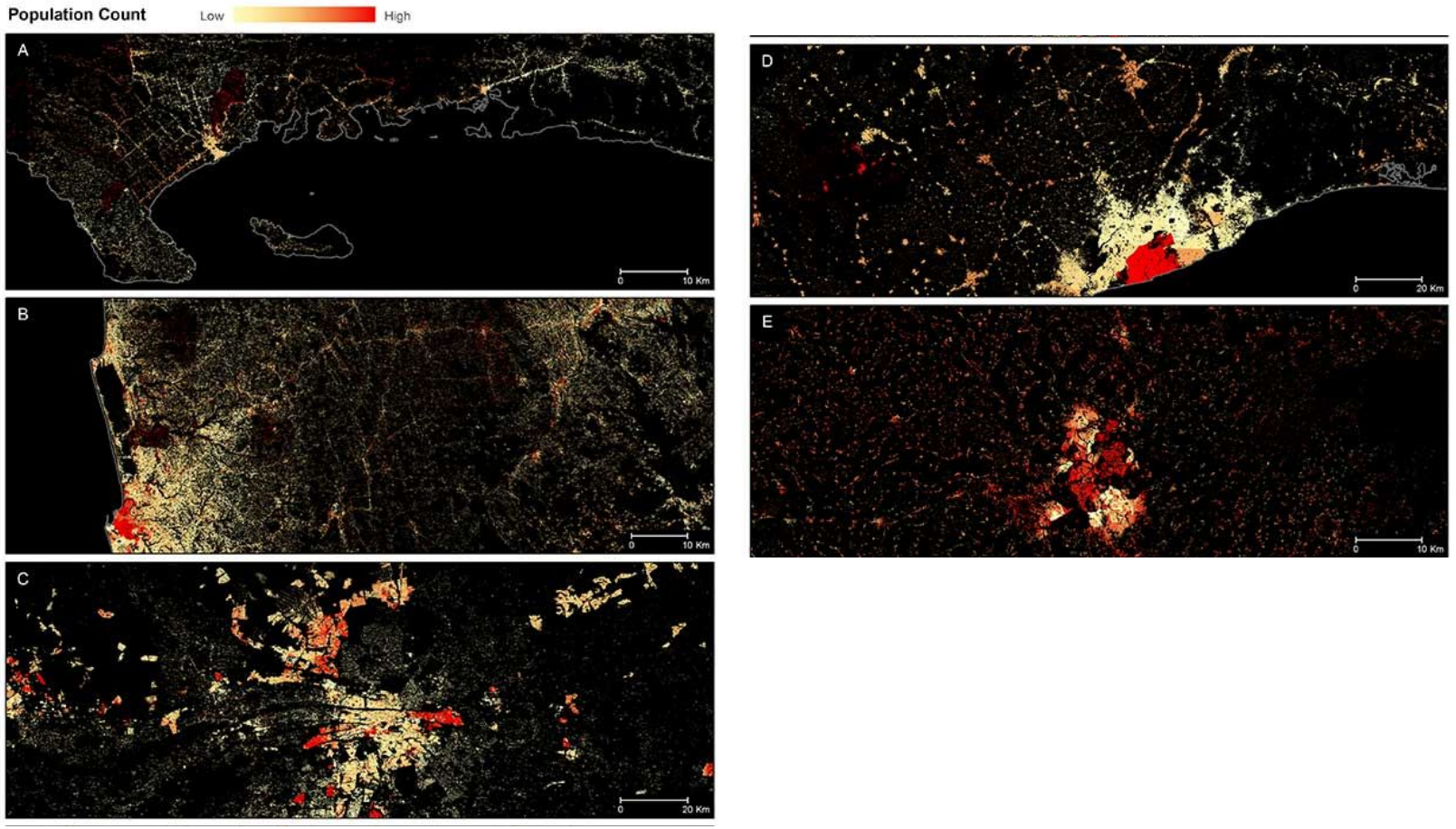


BFA

Segmenting Pop. Density: Columbia Univ. + Facebook

Relevant Examples

- Competitive
- **Academic**
- Catalyst Fund
- FIBR

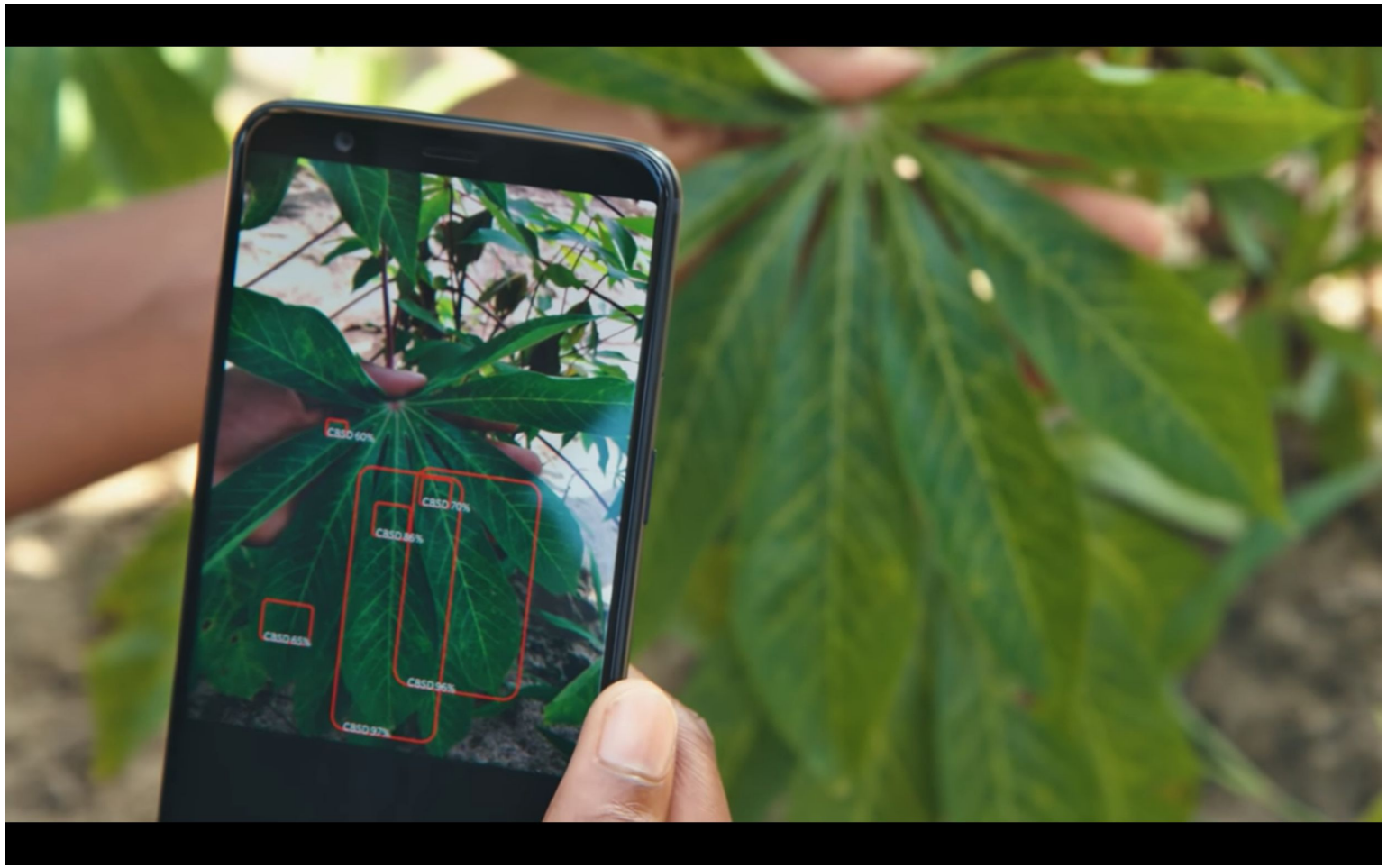


BFA

Relevant Examples

- Competitive
- **Academic**
- Catalyst Fund
- FIBR

Classifying Cassava Health: Penn State Univ. + Google



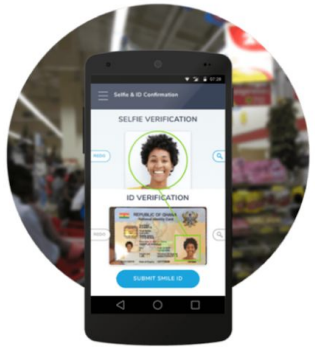
BFA

Relevant Examples

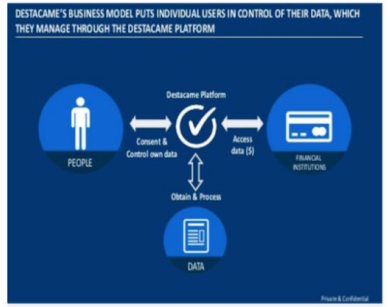
- Competitive
- Academic
- **Catalyst Fund**
- FIBR

Catalyst Fund Examples

Harvesting
Remote Sensing
for input credits



SMILE IDENTITY
Facial Recognition
For Streamlined KYC



DESTACAME
Alternative Data
For Consumer Credit

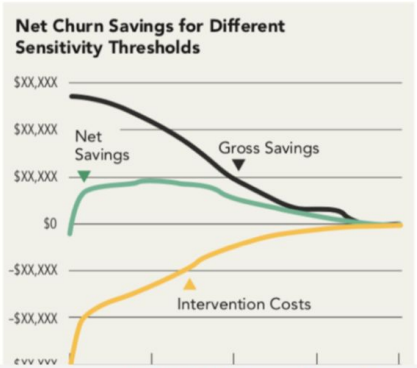
WorldCover
Remote Sensing
for crop insurance

CATALYST FUND

BFA

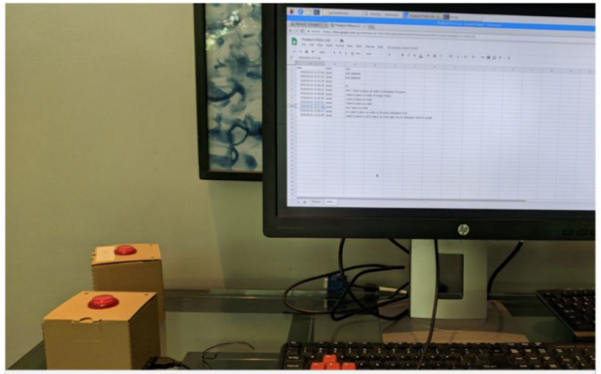
Relevant Examples

- Competitive
- Academic
- Catalyst Fund
- **FIBR**

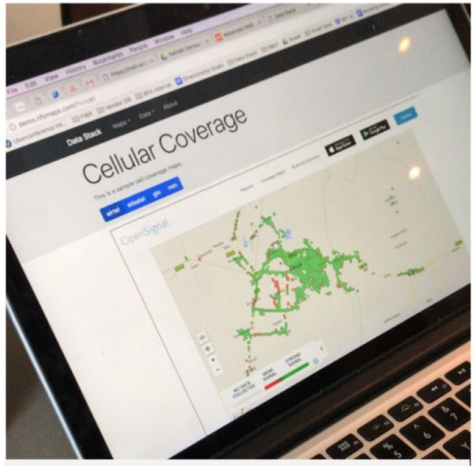


Churn/ROI Prediction

Demand Forecasting



Product Pricing



PAYGo Lead Finder

Inventory Counting

Text to Speech

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

Conclusion
Recap, Q&A, Thoughts

Topics we'll aim to cover in today's bootcamp session...

BFA

MSME Product Counting

- **Right Questions**
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- Creating a Model
- Results

FIBR Nanoproject

(in partnership with Optimetriks)

High Level Questions:

- What do **MSMEs need**, and what specifically do Kenyan Dukas in Kibera need?
- Can we co-create a solution that also solves the **needs of Banks** / MFIs / Lending Groups?

Technical Solution Questions:

- Can we build a **computer vision model** to make analysis of image data from shops **quicker** and **cheaper**?
- How could this **create "practical superpowers"** or **reduce pains** for dukas?

Financial Viability Questions:

- How can current transaction data collected by the partner be leveraged for **sustainable credit provision to shops**?

BFA

MSME Product Counting

- Right Questions
- **Right Actions**
- Assumptions + Hypotheses
- Process + Tools
- Creating a Model
- Results

Identifying Needs:

- Quality sample **images** (assess camera quality, lighting conditions, store display variation, product consistency, etc)
- Reliable **tagging** tool
- Model **training** plan
- Mobile **app that executes model predictions** on non-cutting-edge smartphones, minimize data tx
- Do "**passive** collection" – phone must sit in background to allow shopkeeper to work

MSME Product Counting

- Right Questions
- Right Actions
- **Assumptions + Hypotheses**
- Process + Tools
- Creating a Model
- Results

Hypotheses:

- Digitizing purchases and sales to create a transaction history for the shop will open **new borrowing opportunities**
- CV app will help store owners **increase their sales** by identifying gaps in their product offering
- App can **reduce missed sales** by notifying owners when the stock of a particular item begins to run low
- Manufacturers could **incentivize owners** to more prominently place their products

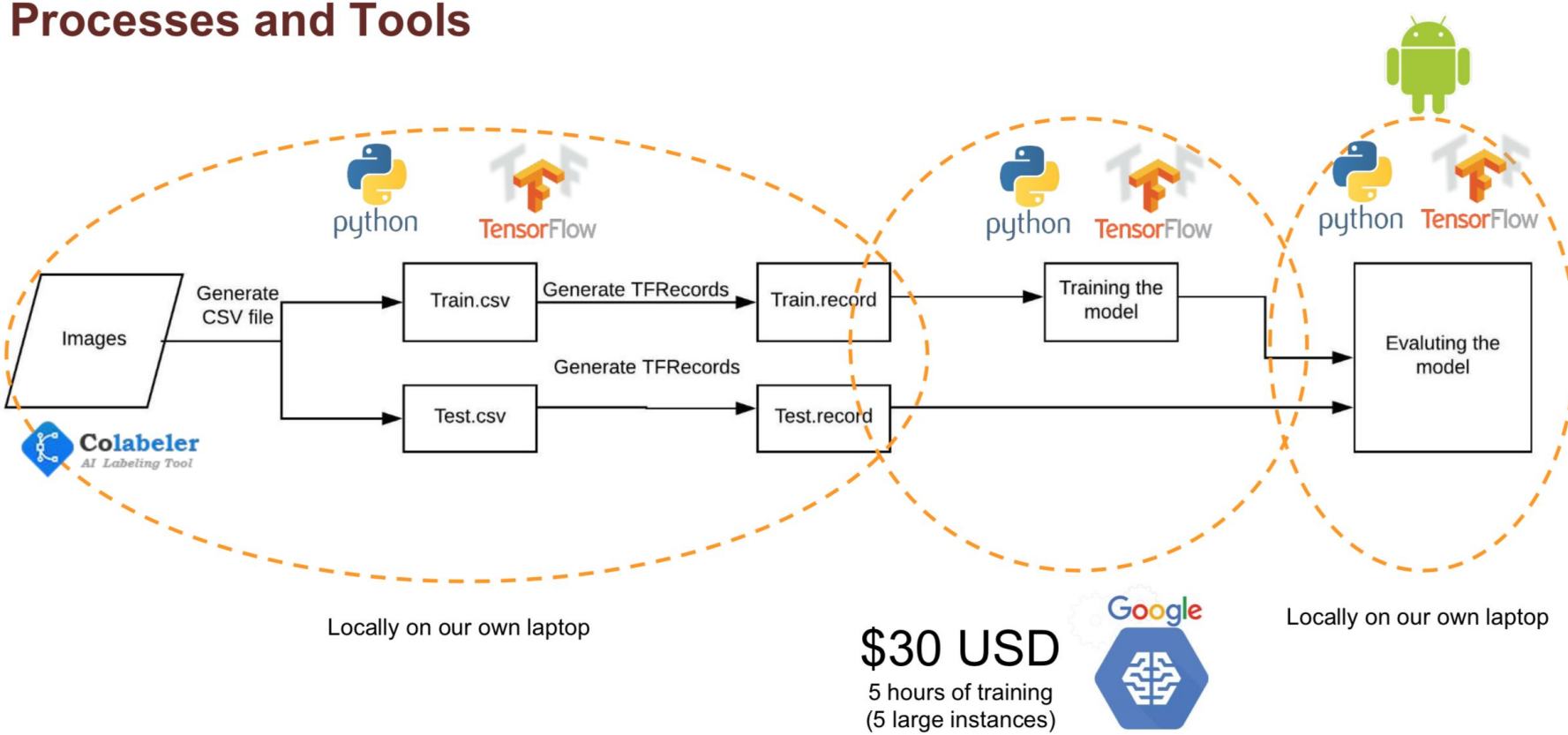
Assumptions:

- Owners **open to advice** on how to better present their product shelves to customers
- Mechanisms exist for automatic or more **automated order fulfillment** based on inventory detected
- Baseline data/knowledge exists that can be used to generate **estimates of inventory** value

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- **Process + Tools**
- Creating a Model
- Results

Processes and Tools

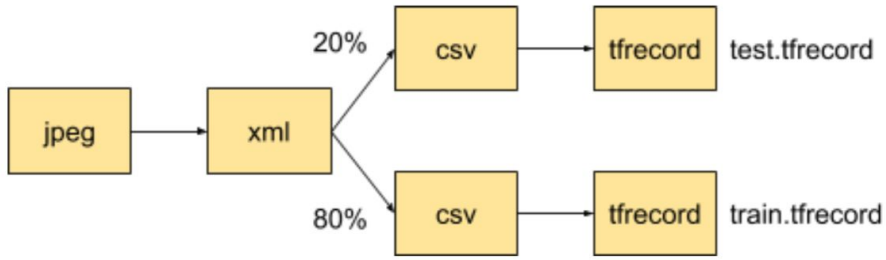


MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- **Process + Tools**
- Creating a Model
- Results

Data Preparation:

- Tagging of Images:
  - Colabeler
  - RectLabel
  - LabelImg
  - LabelBox: now allows direct export to tfrecord



- Splitting into train/test – relatively straightforward
- Conversion into format that model requires
  - Dependent on Model, e.g. Tensorflow requires tfrecord file

Choice:

BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- **Process + Tools**
- Creating a Model
- Results

Framework Selection:

- Looked at ml5js for web, but didn't align with mobile req'ts
- fast.ai was a wonderful option but was evolving rapidly and in beta at the time (i.e. used Keras with Theano, then Keras with TF, now pytorch)
- Ended up using tensorflow directly for good documentation + ease of integration with mobile

Note: this space is still rapidly evolving, so some of this info is already stale!

Choice:

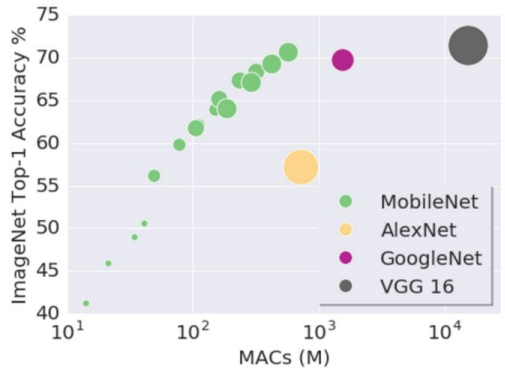
MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- **Process + Tools**
- Creating a Model
- Results

Model Selection:

- For best results, it is recommended to start training from a pre-trained model.
- One of the main ways pre-trained models to vary is how much accuracy has to be traded for speed.
- We selected the **mobilenet v1** model for its speed and stability.
- However, YOLO v3, resnet34, and mobilenet v2 have since been released or updated and may prove to be even faster or higher accuracy.
- After selecting a model, download the checkpoint files to use for training (e.g. model.ckpt for mobilenet v1).

Choice:



BFA

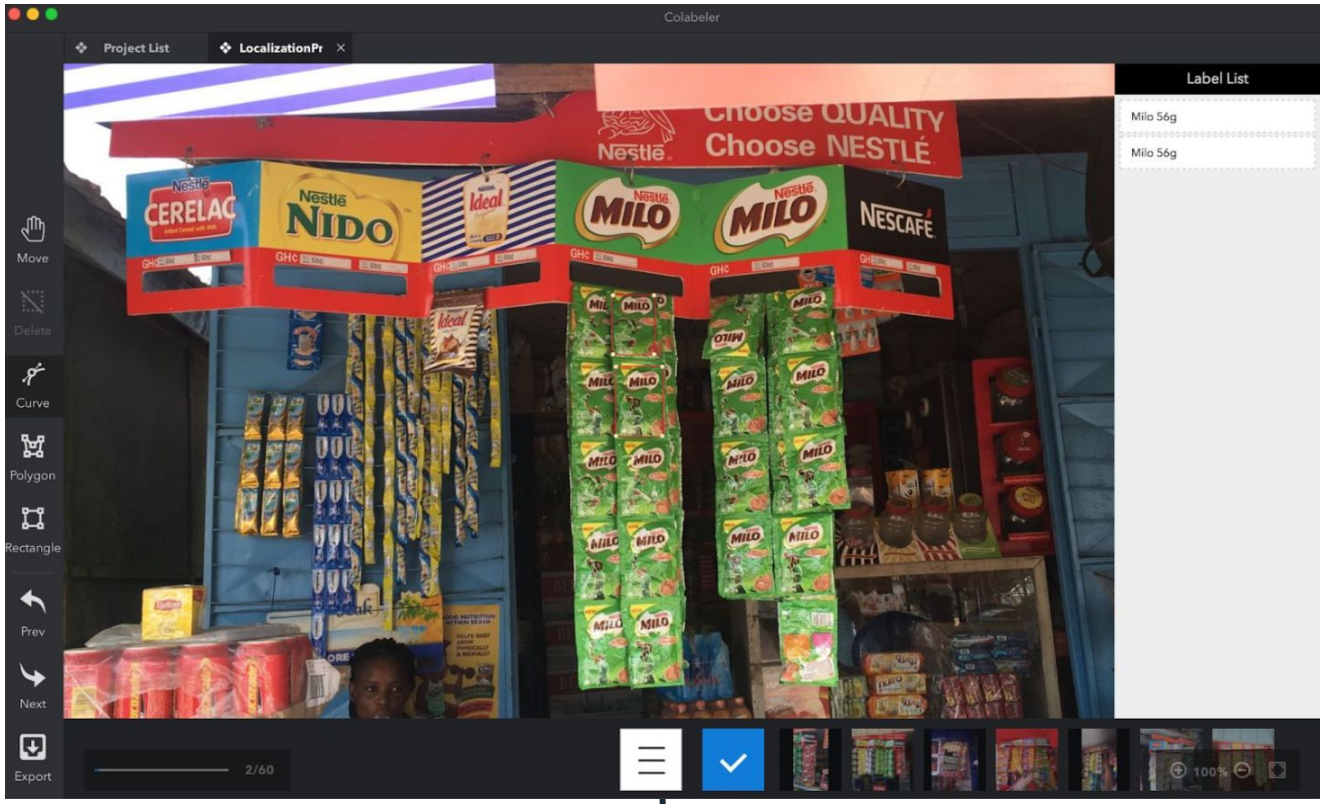
Creating the Model: Preparing Data

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results



export + xml_to_csv.py

filename	width	height	class	xmin	ymin	xmax	ymax
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	357	278	377	394
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	355	400	377	523
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	354	536	390	646
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	445	170	482	209
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	448	210	478	248
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	449	256	476	302
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	450	308	477	352
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	449	357	481	412

BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results

filename	width	height	class	xmin	ymin	xmax	ymax
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	357	278	377	394
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	355	400	377	523
WhatsApp Image 2018-04-26 at 13.47.00 - .jpeg	606	1080	Nescafe-2g	354	536	390	646
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	445	170	482	209
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	448	210	478	248
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	449	256	476	302
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	450	308	477	352
WhatsApp Image 2018-04-26 at 14.00.59 - .jpeg	1024	576	Milo-20g	449	357	481	412

generate_tfrecord.py

```
"""
Usage:
  # From tensorflow/models/
  # Create train data:
  python3 generate_tfrecord.py --csv_input=data/train_labels.cs

  # Create test data:
  python3 generate_tfrecord.py --csv_input=data/test_labels.csv
"""
from __future__ import division
from __future__ import print_function
from __future__ import absolute_import

import os
import io
import pandas as pd
import tensorflow as tf

from PIL import Image
from object_detection.utils import dataset_util
from collections import namedtuple, OrderedDict

flags = tf.app.flags
flags.DEFINE_string('csv_input', '', 'Path to the CSV input')
flags.DEFINE_string('output_path', '', 'Path to output TFRecor
FLAGS = flags.FLAGS


# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'Cerelac Yellow Can':
        return 21
    if row_label == 'Cerelac Pink Can':
```

pbtxt file

```
item {
 id: 1
 name: 'Ideal-Brown-37g'
}
item {
 id: 2
 name: 'Ideal-Blue-37g'
}
item {
 id: 3
 name: 'Nescafe-2g'
}
item {
 id: 4
 name: 'Nescafe-36g'
}
item {
 id: 5
 name: 'Milo-20g'
}
```

TFRecord

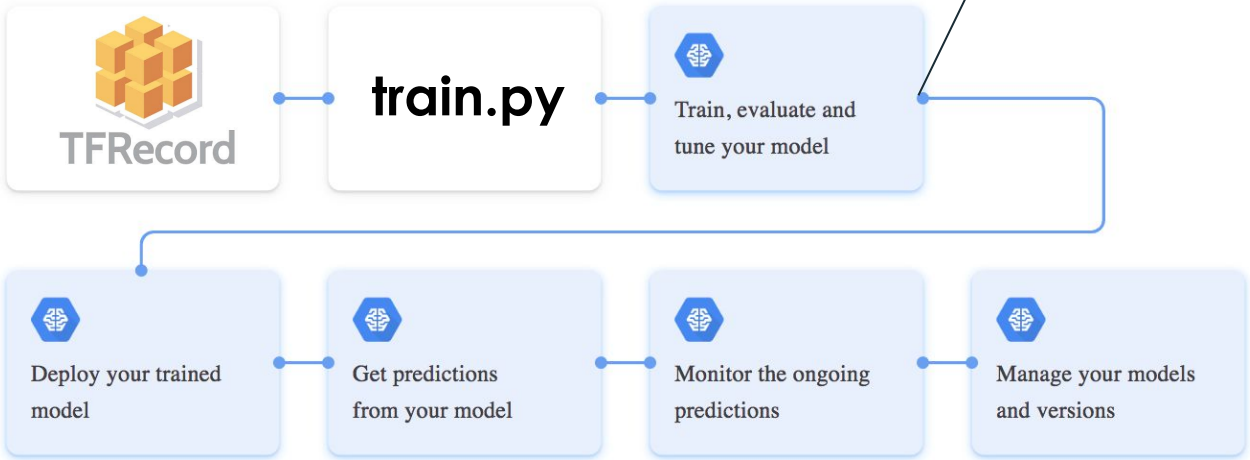
BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results

Training the Model: Local vs Cloud

-or-



Note: Monitor training via TensorBoard

BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results

Prepping the Model: Converting

Checkpoints from training

```
model.ckpt-1.data-00000-of-00001
model.ckpt-1.index
model.ckpt-1.meta
model.ckpt-200.data-00000-of-00001
model.ckpt-200.index
model.ckpt-200.meta
```

export_inference_graph.py

```
python export_inference_graph \
    --input_type image_tensor \
    --pipeline_config_path path/to/ssd_inception_v2.config \
    --trained_checkpoint_prefix path/to/model.ckpt \
    --output_directory path/to/exported_model_directory \
    --config_override " \
            model{ \
              faster_rcnn { \
                second_stage_post_processing { \
                  batch_non_max_suppression { \
                    score_threshold: 0.5 \
                  } \
                } \
              } \
            }"
"""
import tensorflow as tf
from google.protobuf import text_format
from object_detection import exporter
from object_detection.protos import pipeline_pb2

slim = tf.contrib.slim
flags = tf.app.flags

flags.DEFINE_string('input_type', 'image_tensor', 'Type of input node. Can be '
                    'one of [`image_tensor`, `encoded_image_string_tensor`, '
                    '`tf_example`]')
flags.DEFINE_string('input_shape', None,
                    'If input_type is `image_tensor`, this can explicitly set '
                    'the shape of this input tensor to a fixed size. The '
                    'dimensions are to be provided as a comma-separated list '
                    'of integers. A value of -1 can be used for unknown '
                    'dimensions. If not specified, for an `image_tensor, the '
                    'default shape will be partially specified as '
                    '`[None, None, None, 3]`.')
flags.DEFINE_string('pipeline_config_path', None,
                    'Path to a pipeline_pb2.TrainEvalPipelineConfig config '
```

frozen_inference_graph.pb (!!!)

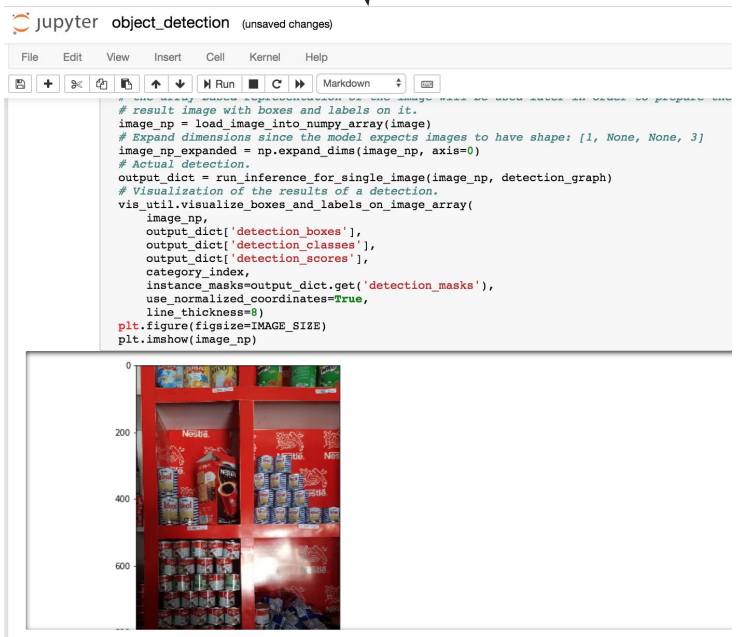
BFA

Running the Model: Notebook vs App

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- **Creating the Model**
- Results

frozen_inference_graph.pb + Test Data



frozen_inference_graph.pb + label.txt

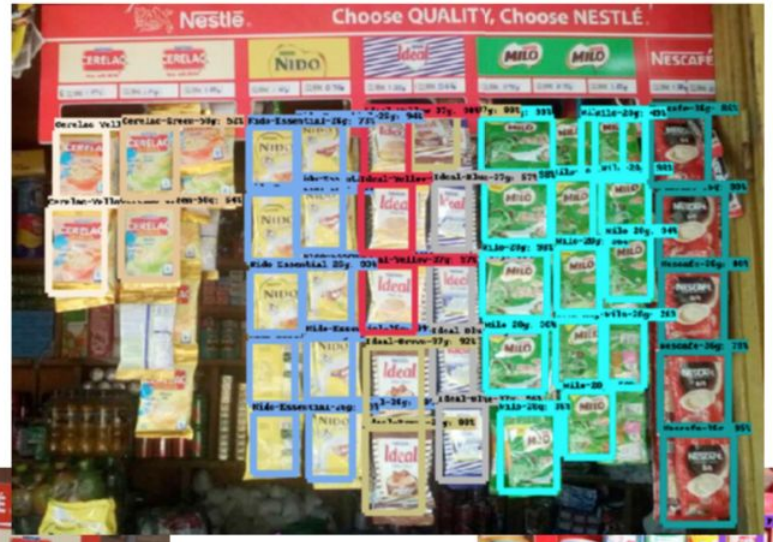
TensorFlow Demo App /assets [in DetectorActivity.java, update the paths TF_OD_API_MODEL_FILE and TF_OD_API_LABELS_FILE]

BFA

Intended Results: Success!

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- Creating a Model
- **Results**



BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- Creating a Model
- **Results**

Unintended Results:
TODOs for Next Iteration!

performs poorly with overlapped, blurry and side views

BFA

MSME Product Counting

- Right Questions
- Right Actions
- Assumptions + Hypotheses
- Process + Tools
- Creating a Model
- **Results**

Unintended Results:
TODOs for Next Iteration!

BFA

What did we learn from this?

- Computer vision is no longer a technology that requires a team of researchers and data scientists to implement.
- Training a model is a multi-step process that can be error-prone due to the myriad small details that are required.
- Selecting libraries and tools that have a strong community and ample documentation is critical to being able to troubleshoot issues that arise during implementation.
- Labeled datasets containing as few as a hundred labels per product is sufficient for producing a model with relatively high accuracy.

What did we learn from this?

- There are many less-than-ideal scenarios that exist in the real world that can affect the accuracy of the model such as products being blocked or obscured in the images.
- Cloud computing can be leveraged to save time and resources for an additional cost.
- Trained models can easily be run in mobile apps to enable offline use, but keeping these apps up-to-date would require version control or app updates.

Overview
What is DL?

Zooming In
Computer Vision + Architectures

CV Examples
Looking at Applications

CV + MSMEs
Case study of product counting

Conclusion
Recap, Q&A, Thoughts

Topics we'll aim to cover in today's bootcamp session...

BFA

Recap from today's session:

- Deep Learning is made up of relatively simple components
- Many major academic and tech players are already leveraging these tools to augment existing solutions
- There is an appetite from donors and investors to explore pilots
- Opportunities exist for Deep Learning to have concrete impact!
- By sticking to a framework (e.g. proposed in FIBR's AI Report), we can hold technological solutions accountable to the real needs

Thank you! Questions?

Matt Grasser | @mr_z_ro
Director of Inclusive Fintech, BFA
https://www.linkedin.com/in/msgrasser