TENSORFLOW

Leveraging TensorFlow for Locating FTC Game Pieces

October 19, 2019, v03
cc Dr. Edward C. Epp
Oregon IEEE RAS Chair
edward.c.epp@ieee.org

Tensorflow

- Google
- Library for Machine Learning
- Can be used to create Convolution Neural Networks
- TensorFlow Lite is for mobile platforms
- Open Source

Caffe

- Deep learning framework by Berkeley (Convolutional Architecture for Fast Feature Embedding)_

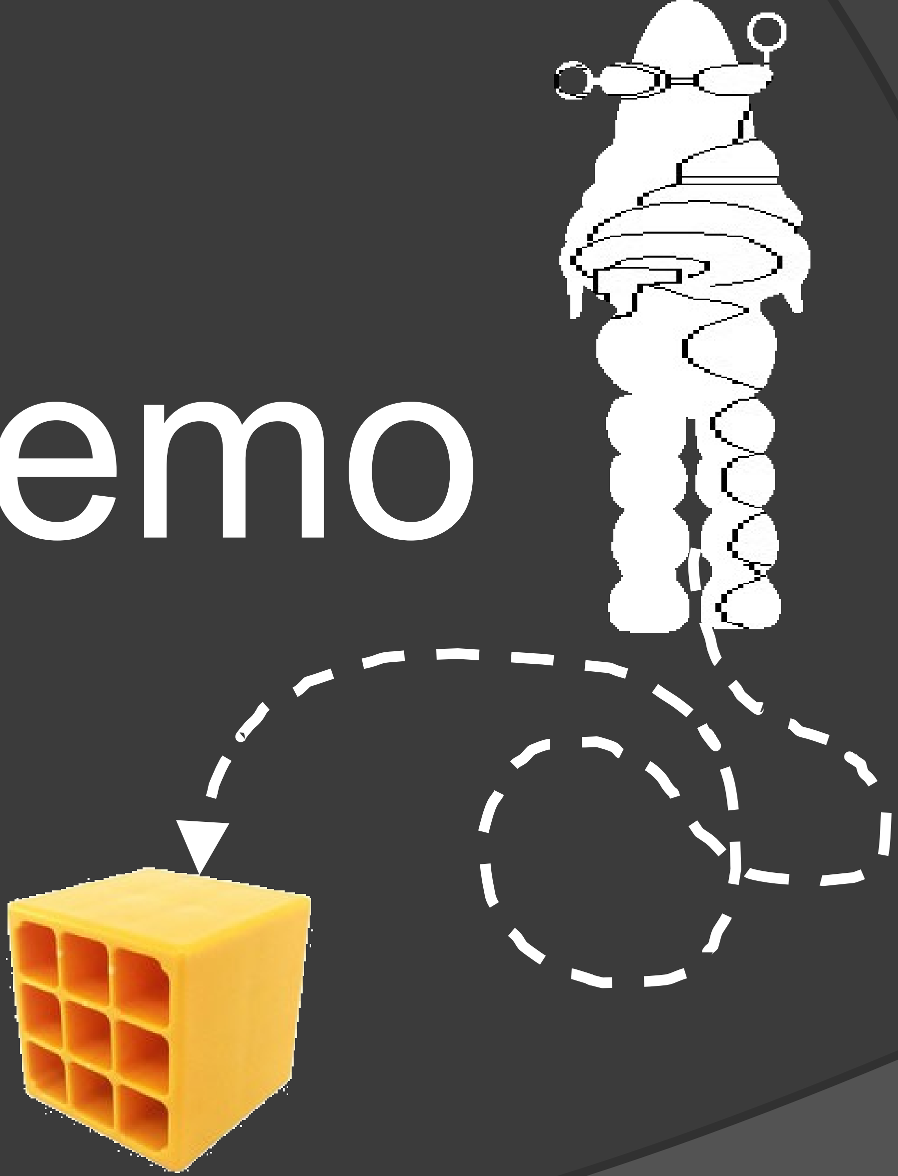
Outline

- Demo: Lets see machine learning in action
  - Targeting a game piece
- Code walkthrough: How it is implemented
  - Enhancing the code
- TensorFlow confidence and shortcomings
- Building blocks: How it does it
  - Convolutions
  - Neural Networks

Target Audience

- Skilled in basic autonomous programming
  - Upgrade to new SDKs
  - Copy sample programs into TeamCode
  - Set hardware configuration
  - Set up direct communications between driver station and robot controller
- Java skills
  - Basic selection (if and switch)
  - Basic iteration (while)
  - Basic class structure
  - Basic function calls

Demo



Target and Push the Gold Mineral

- <video as backup>

Code Walkthrough

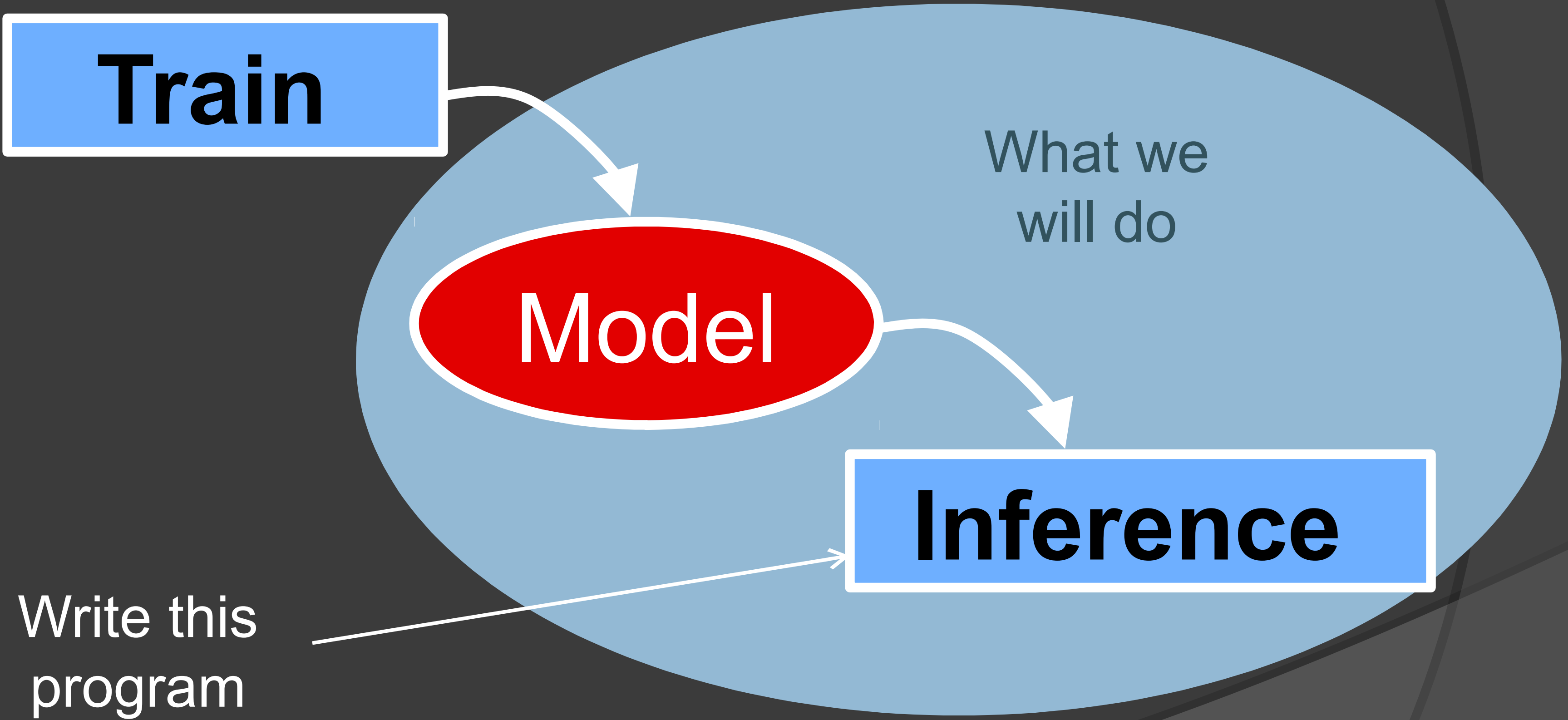
https://www.iconfinder.com/icons/347028/hero_intangibility_mutant_teleportation_walk_through_warp_icon

I Learn by Doing

- Experimenting with demos
- Exploring modifications
- Analyzing things that don’t work
- Finally, reading About things I don’t understand

https://deathstar.bandcamp.com/track/broken-robots-feat-kyle-stevens

Convolutional Neural Networks CNN



Train

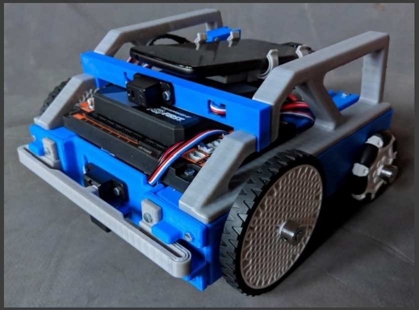
Model

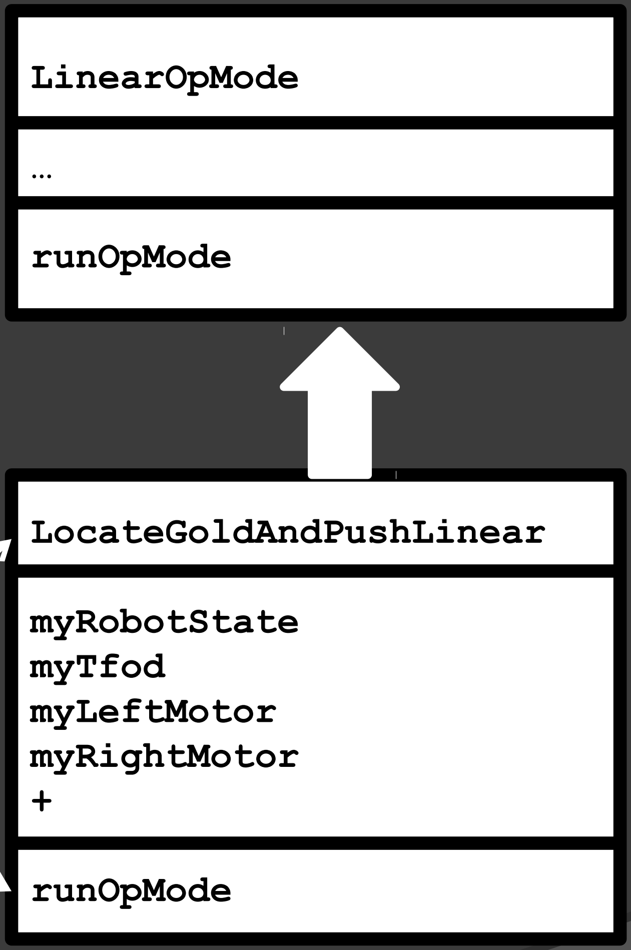
Inference

What we will do

Write this program

Object Oriented Programming





LinearOpMode

…

runOpMode

Object inheritance

LocateGoldAndPushLinear

myRobotState
myTfod
myLeftMotor
myRightMotor
+

runOpMode

Object name

Object state

Object operations

FTC Robot Controller

Create object

Run operation

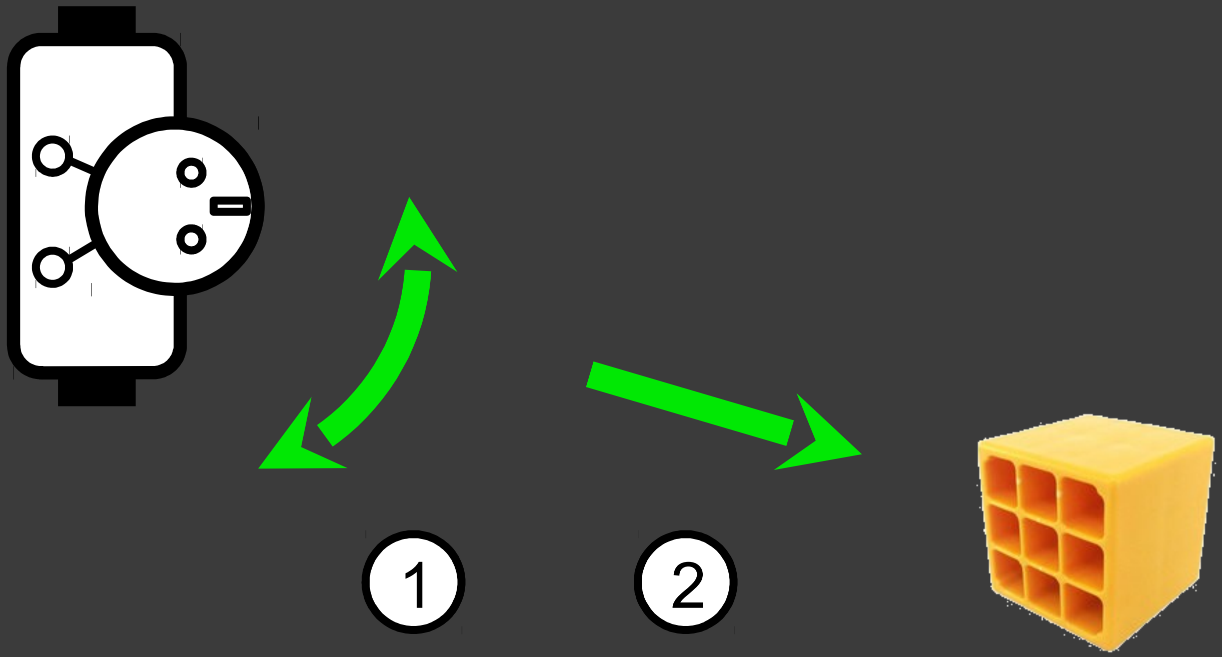
High Level Organization

```
@Autonomous(name = "Locate Gold And Push Linear",
            group = "Concept”)
public class LocateGoldAndPushLinear extends LinearOpMode
{
    // constants
    private static final int SCREEN_WIDTH        = 1280;

    // Robot state - The Tensor Flow Object Detection engine.
    private RobotState myRobotState = TARGET_GOLD;
    private TFObjectDetector myTfod = null;

    Public void runOpMode () {…}

    // Helper functions
    Public void initRobot() {…}
    private void moveToGold(){…}
}
```

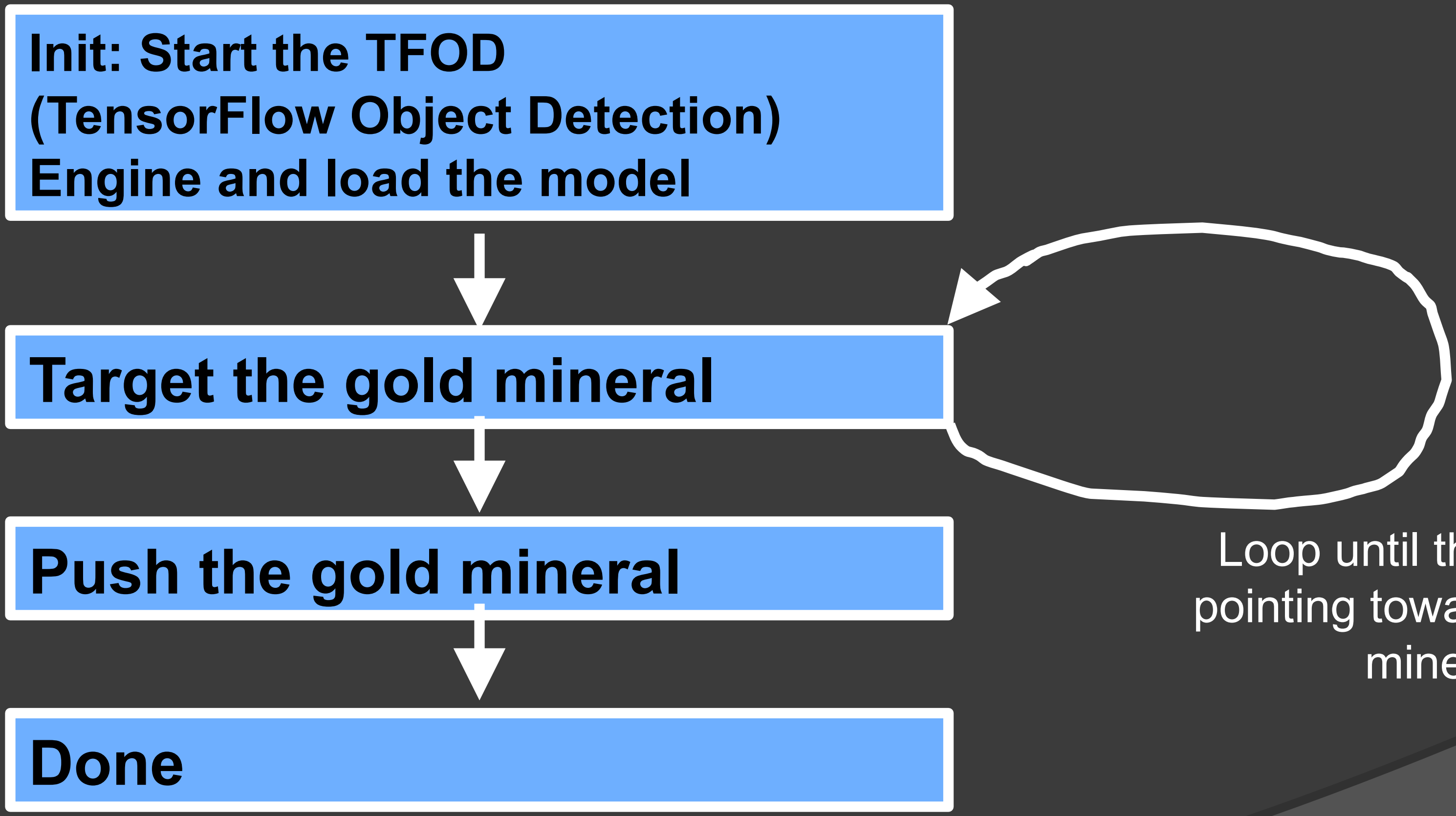
Code Start

```
public void runOpMode()
{
    initRobot();

    /** Wait for the game to begin */
    telemetry.addData(">", "Press Play to start tracking");
    telemetry.update();
    waitForStart();

    // Main Linear OpMod loop
    while (opModeIsActive())
    {
        // State machine to control the robot
    }
}
```

Locate and Push



State Diagram



Init: Start the TFOD (TensorFlow Object Detection) Engine and load the model

Target the gold mineral

Push the gold mineral

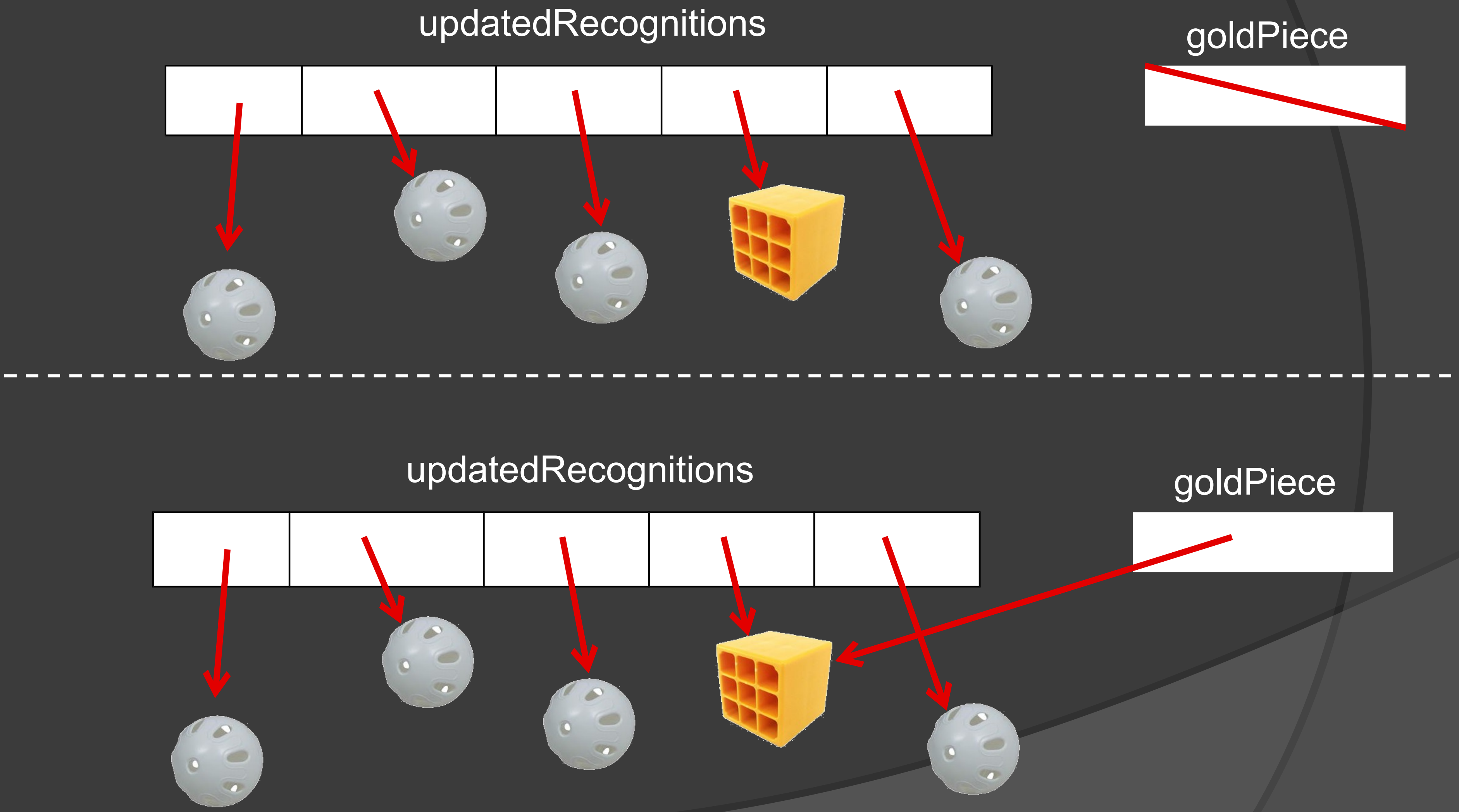
Done

Loop until the robot is pointing toward the gold mineral

State Machine to Control Robot

```
switch (myRobotState)
{
    case TARGET_GOLD:
        targetGold();
        break;
    case MOVE_GOLD:
        moveToGold();
        break;
    case DONE:
        shutdown();
        break;
    case TEST:
        moveFor(200,200);
        moveFor(-200, 200);
        myRobotState = DONE;
        break;
    case ERROR:
        myRobotState = DONE;
        break;
    default:
        myRobotState = ERROR;
}
```

Finding the Gold



updatedRecognitions

goldPiece

updatedRecognitions

goldPiece

targetGold Outline

```
private void targetGold ()
{
    Recognition goldPiece = null;
    List<Recognition> updatedRecognitions =
                               myTfod.getUpdatedRecognitions();

    if (updatedRecognitions != null)
    {
        for (Recognition recognition : updatedRecognitions)
        {
            if (recognition.getLabel().equals(LABEL_GOLD_MINERAL))
            {
                goldPiece = recognition;
                break;
            }
        }
        if (goldPiece != null) {     // missing an error check?
        {
            // Found gold – face toward it – see next page
        }
    }
    else
    {
        idle();
    }
}
```

Recognition Operations

```
Recognition recognition = …;
if (recognition.getLabel().equals(LABEL_GOLD_MINERAL))
  int goldMineralLeftX  = (int) goldPiece.getLeft();
```

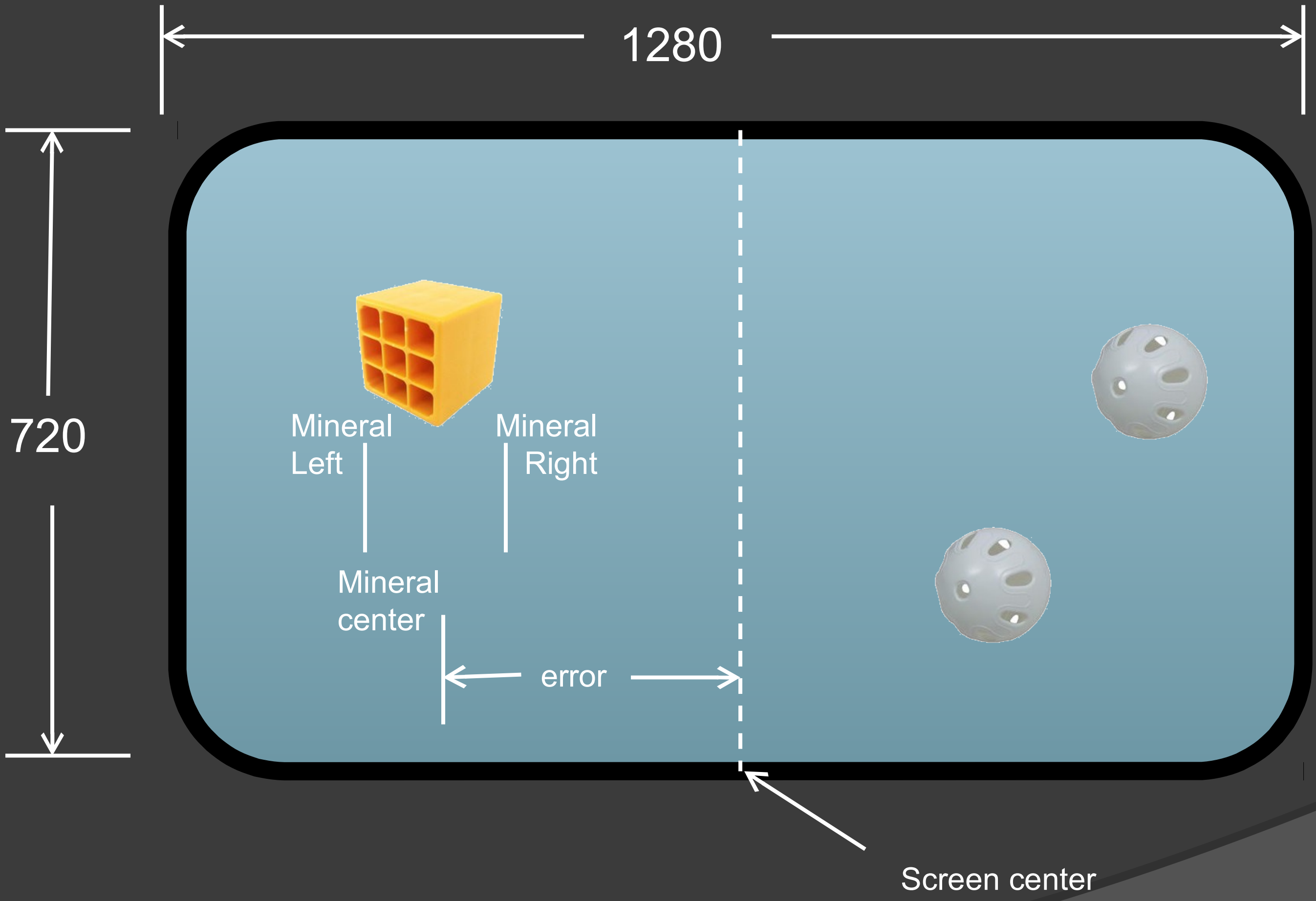
List of operations

```
    estimateAngleToObject (AngleUnit angleUnit) // DEGREE or RADIANS
    getBotton ()           // bounding box
    getConfidence ()
    getHeight ()           // bounding box
    getImageHeight ()      // Entire image
    getImageWidth ()       // Entire image
    getLable ()
    getLeft ()             // bounding box
    getRight ()            // bounding box
    getTop ()              // bounding box
    getWidth ()            // bounding box
```

Finding the Recognition API

- Download Skystone repository if you haven’t
  - https://github.com/FIRST-Tech-Challenge/SkyStone
- Unzip Skystone-master.zip
- cd …/SkyStone-master/doc/javadoc
- Open allclasses-frame.html in web browser
- Click on the “Recognition” link

Cell Phone Screen



targetGold Found Gold

*** find the error ***

```
if (goldPiece != null)
{
    int goldMineralLeftX  = (int) goldPiece.getLeft();
    int goldMineralRightX = (int) goldPiece.getLeft();
    int goldMineralCenterX =
       (goldMineralLeftX + goldMineralRightX) / 2;
    int error = goldMineralCenterX - SCREEN_WIDTH / 2;
    if (Math.abs(error) < POINTING_TOLERANCE)
    {
        myRobotState = MOVE_GOLD;
    }
    else
    {
        int turn_clicks = error / 8;
        moveFor(turn_clicks, -turn_clicks);
    }
}
```

moveFor

```
// if leftCount  is < the rightCount the robot will turn left
// if rightCount is < the leftCount the robot will turn right

private void moveFor (int leftCount, int rightCount)
{
    myLeftMotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    myRightMotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    myLeftMotor.setTargetPosition(leftCount);
    myRightMotor.setTargetPosition(rightCount);
    myLeftMotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    myRightMotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);
    myLeftMotor.setPower(MID_SPEED);
    myRightMotor.setPower(MID_SPEED);
    while (opModeIsActive() &&
              (myLeftMotor.isBusy() || myRightMotor.isBusy()))
    {
        idle();
    }
}
```

moveToGold

```
private void moveToGold()
{
    moveFor(CLICKS_TO_TARGET, CLICKS_TO_TARGET);
    myRobotState = DONE;
}
```

Code Modifications

How would you modify the previous code snippet to locate a gold piece not in the current view?

1. Do it in two discrete steps.
   a. Turn
   b. Check
2. Do it in one sweeping motion
   a. Turn and check simultaneously
   b. What will the challenge be?

targetGold Outline (duplicate)

```
private void targetGold ()
{
    Recognition goldPiece = null;
    List<Recognition> updatedRecognitions =
                              myTfod.getUpdatedRecognitions();

    if (updatedRecognitions != null)
    {
        for (Recognition recognition : updatedRecognitions)
        {
            if (recognition.getLabel().equals(LABEL_GOLD_MINERAL))
            {
                goldPiece = recognition;
                break;
            }
        }
        if (goldPiece != null) {
        {
            // Found gold – face toward it – see next page
        }
    }
    else
    {
        idle();
    }
}
```

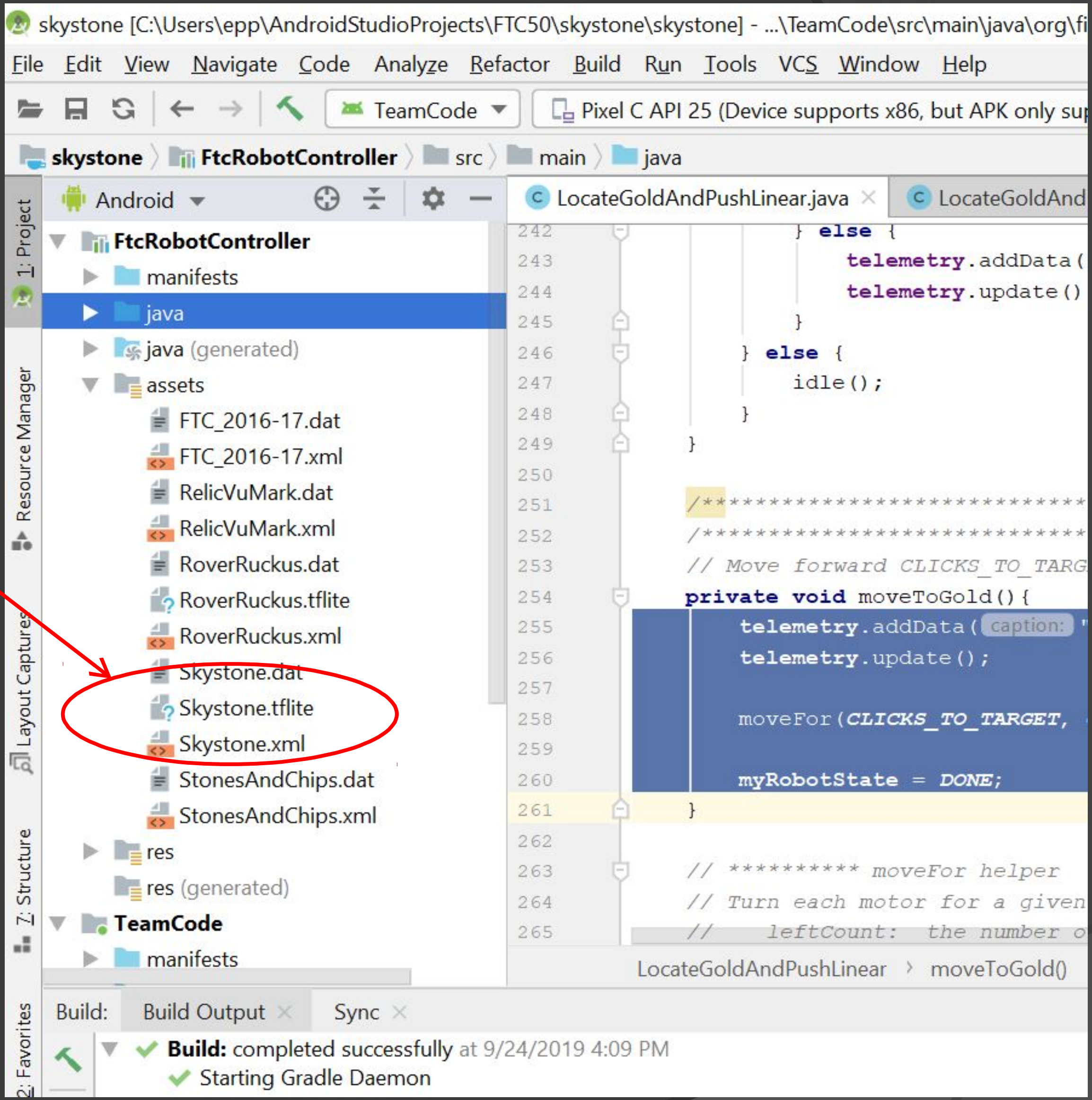
initRobot

```
private void initRobot()
{
    initVuforia();

    if (myRobotState != ERROR)
    {
        initTfod();
    }

    if (myRobotState != ERROR)
    {
        initMotors();
    }
}
```

initTfod   TensorFlow Object Detection

```
private static final String TFOD_MODEL_ASSET     = "RoverRuckus.tflite";

private void initTfod()
{
    if (ClassFactory.getInstance().canCreateTFObjectDetector())
    {
        int tfodMonitorViewId = hardwareMap.appContext.getResources().getIdentifier(
                "tfodMonitorViewId", "id”, hardwareMap.appContext.getPackageName());
        TFObjectDetector.Parameters tfodParameters =
             new TFObjectDetector.Parameters(tfodMonitorViewId);
        myTfod = ClassFactory.getInstance().createTFObjectDetector(
             tfodParameters, myVuforia);
        myTfod.loadModelFromAsset(TFOD_MODEL_ASSET,
                                  LABEL_GOLD_MINERAL, LABEL_SILVER_MINERAL);
        if (myTfod != null)
        {
            myTfod.activate();
        }
        else
        {
            telemetry.addData("ERROR", "TensorFlow lite did not activate");
            myRobotState = ERROR;
        }
    }
    else
    {
        telemetry.addData("ERROR", "This device is not compatible with TFOD");
        myRobotState = ERROR;
    }
}
```

Model location in SDK



- FtcRobotController
  - manifests
  - java
  - java (generated)
  - assets
    - FTC_2016-17.dat
    - FTC_2016-17.xml
    - RelicVuMark.dat
    - RelicVuMark.xml
    - RoverRuckus.dat
    - RoverRuckus.tflite
    - RoverRuckus.xml
    - Skystone.dat
    - Skystone.tflite
    - Skystone.xml
    - StonesAndChips.dat
    - StonesAndChips.xml
  - res
  - res (generated)
- TeamCode
  - manifests

```
                } else {
                    telemetry.addData(
                    telemetry.update()
                }
            } else {
                idle();
            }
        }

        /***************************
        /***************************
        // Move forward CLICKS_TO_TARG
        private void moveToGold(){
            telemetry.addData( caption: "
            telemetry.update();

            moveFor(CLICKS_TO_TARGET, 

            myRobotState = DONE;
        }

        // ********** moveFor helper
        // Turn each motor for a given
        //     leftCount:  the number o
```

```
Build: completed successfully at 9/24/2019 4:09 PM
  Starting Gradle Daemon
```

https://github.com/ftctechnh/ftc_app/tree/master/FtcRobotController/src/main/assets

Exercises 1

https://www.pinterest.com/loveverything6087/

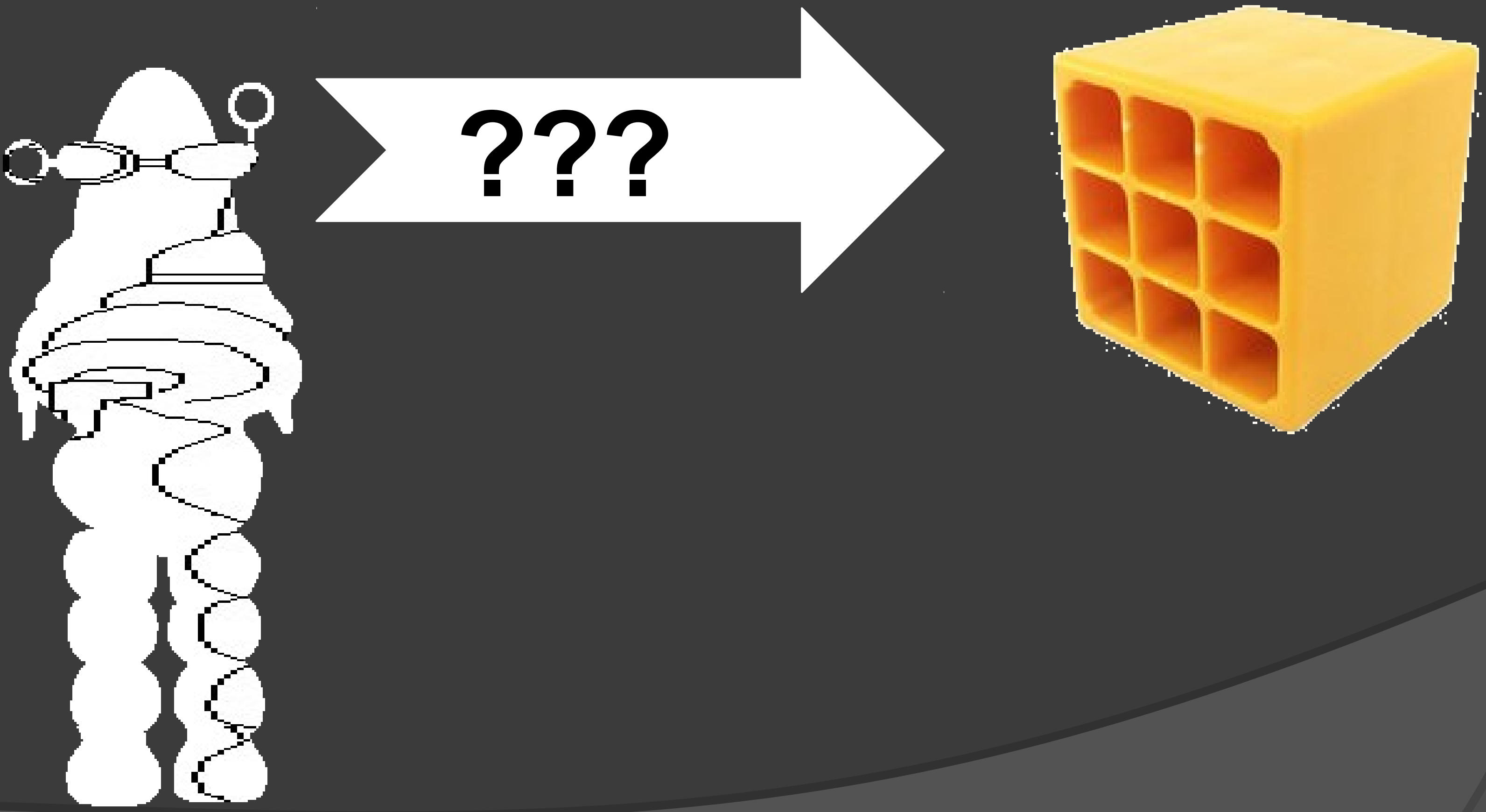
Agile Programing

1. Identify a simple problem subset
2. Develop a solution
3. Test
4. Analyze
5. Identify a simple enhancement
6. Go to step 2

Exercises 1 - 4

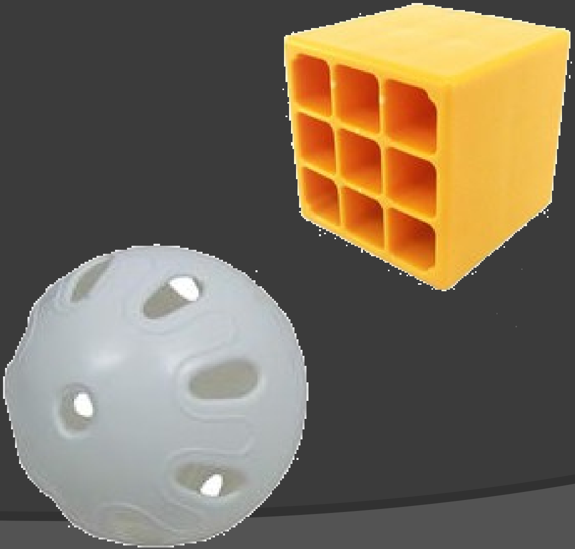
- Build and test the demo program
  - LocateGoldAndPushLinear.java
- Modify the demo program to hunt for gold not visible in display
- Modify the demo program to work with Skystone objects
- Go after silver silver objects after pushing the gold mineral

Confidence
How well does it work?

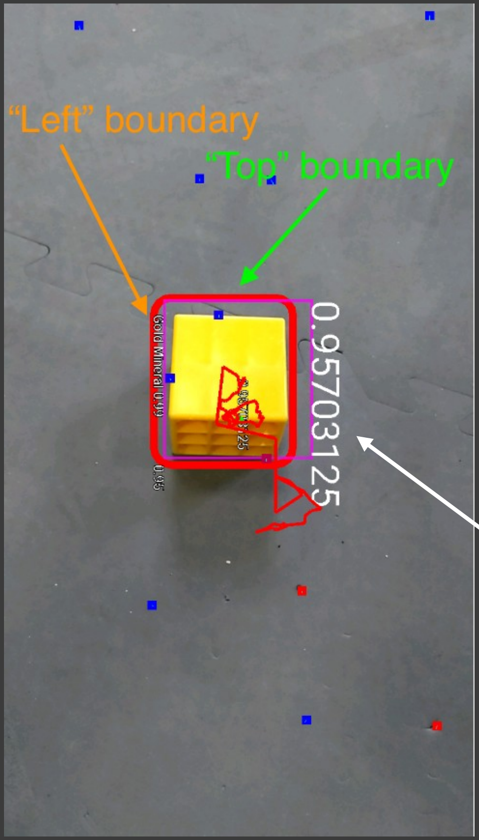


Model Training Constraints Impacts Testing

- A specific game environment (arena)
  - Varnished wood appear like a gold piece
  - Glare off a wood floor without tiles
  - Ambient light not controlled
- Trained to recognize two game pieces
  - Not trained that a yellow car is not a gold piece but rather a toy car
  - Wicker basket resembles a gold piece



TensorFlowConfidence Levels



- The minimum detection confidence level is set to 40%.
- Confidence level of 40% or higher is required in order to consider an object as being detected in its field of view.

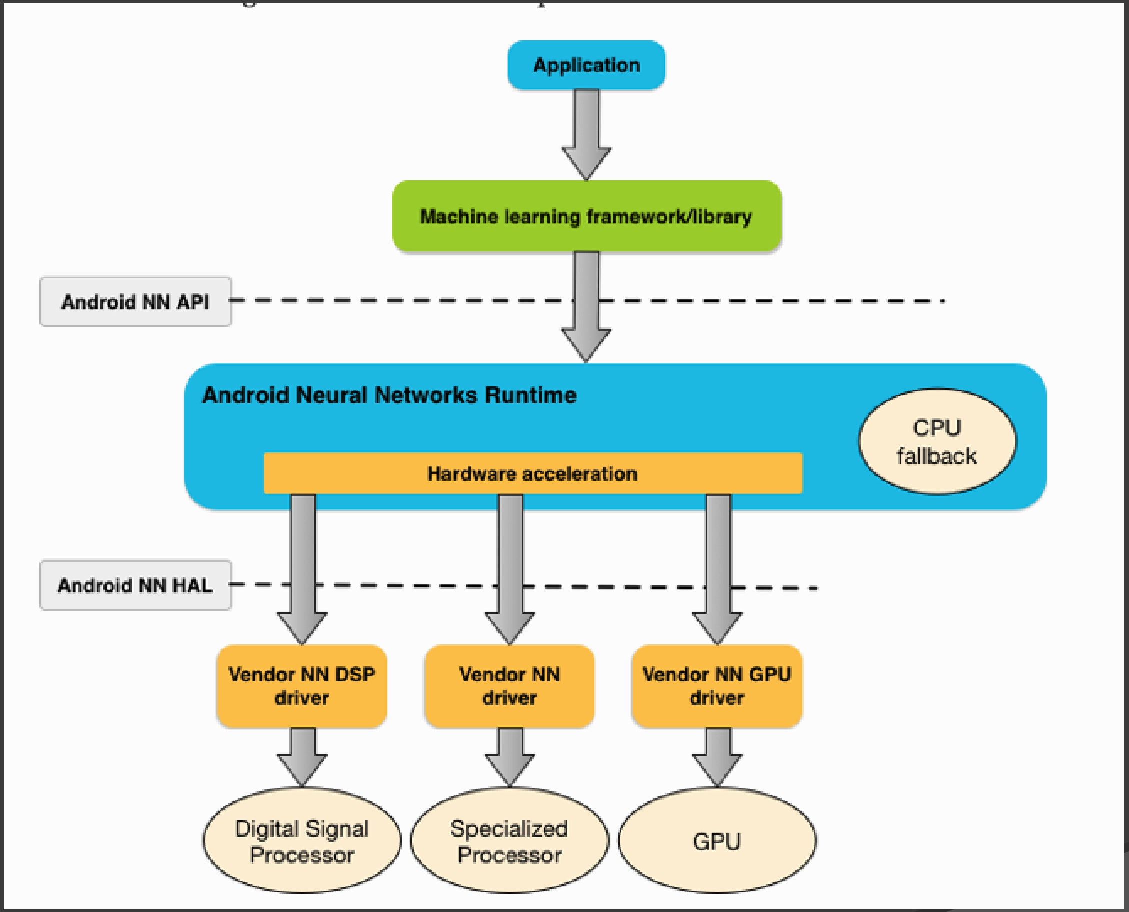
Confidence value

https://github.com/ftctechnh/ftc_app/wiki/Java-Sample-TensorFlow-Object-Detection-Op-Mode

FTC TensorFlow Lite Shortcomings

- Recognition confidence is not 100%
- Training is targeted at FTC game
- Lighting is critical
- Objects may be lost in surrounding clutter
- Angle of the object may be important
- There is a significant distance penalty
- TensorFlow requires compute overhead
  - It requires a more capable Robot Control phone
- Good programming practices are important
  - Stay tuned to the forum

TensorFlow Lite Inference Engine



https://quantumcomputingtech.blogspot.com/2019/01/android-machine-learning-tutorial.html

On the Forum

Training is Hard

“… it would be great if the TensorFlow model could be trained with some of the training images taken with the stones/skystones as they appear in their row in the quarry from varying distances (and not just as "loose" separated stones) ...”

Cheer4FTC

“However, the time consuming part is generating and selecting the training images. We (an intern and I) spent several weeks generating different training data sets, and developing models based on what we thought were ideal training sets. However, in some cases, adding additional types of images, in an attempt to increase the detection accuracy, resulted in worse results.”

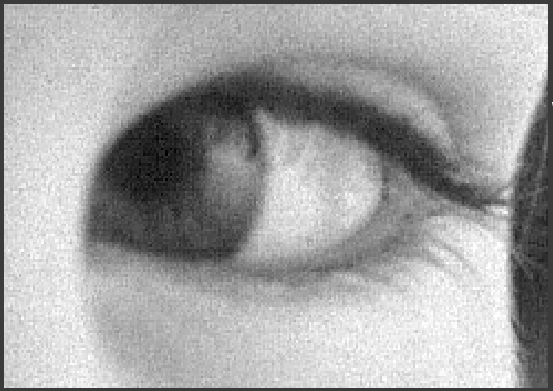
Tom Eng

https://ftcforum.usfirst.org/forum/ftc-technology/blocks-programming/74443-sample-op-mode-and-video-on-navigating-to-a-skystone-during-the-autonomous-period

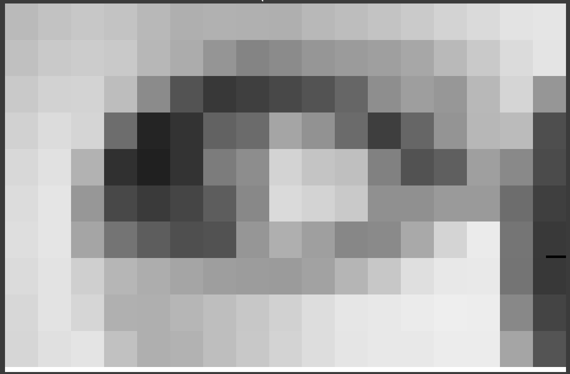
How it Works

https://say-digital.io/how-it-works/

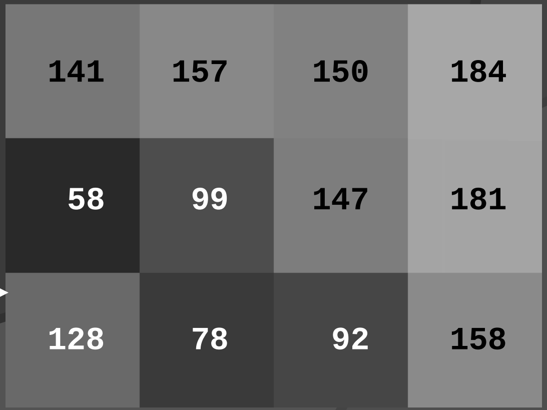
Camera Image



A scene transformed
into pixels



4 x 3 region of
the above Image

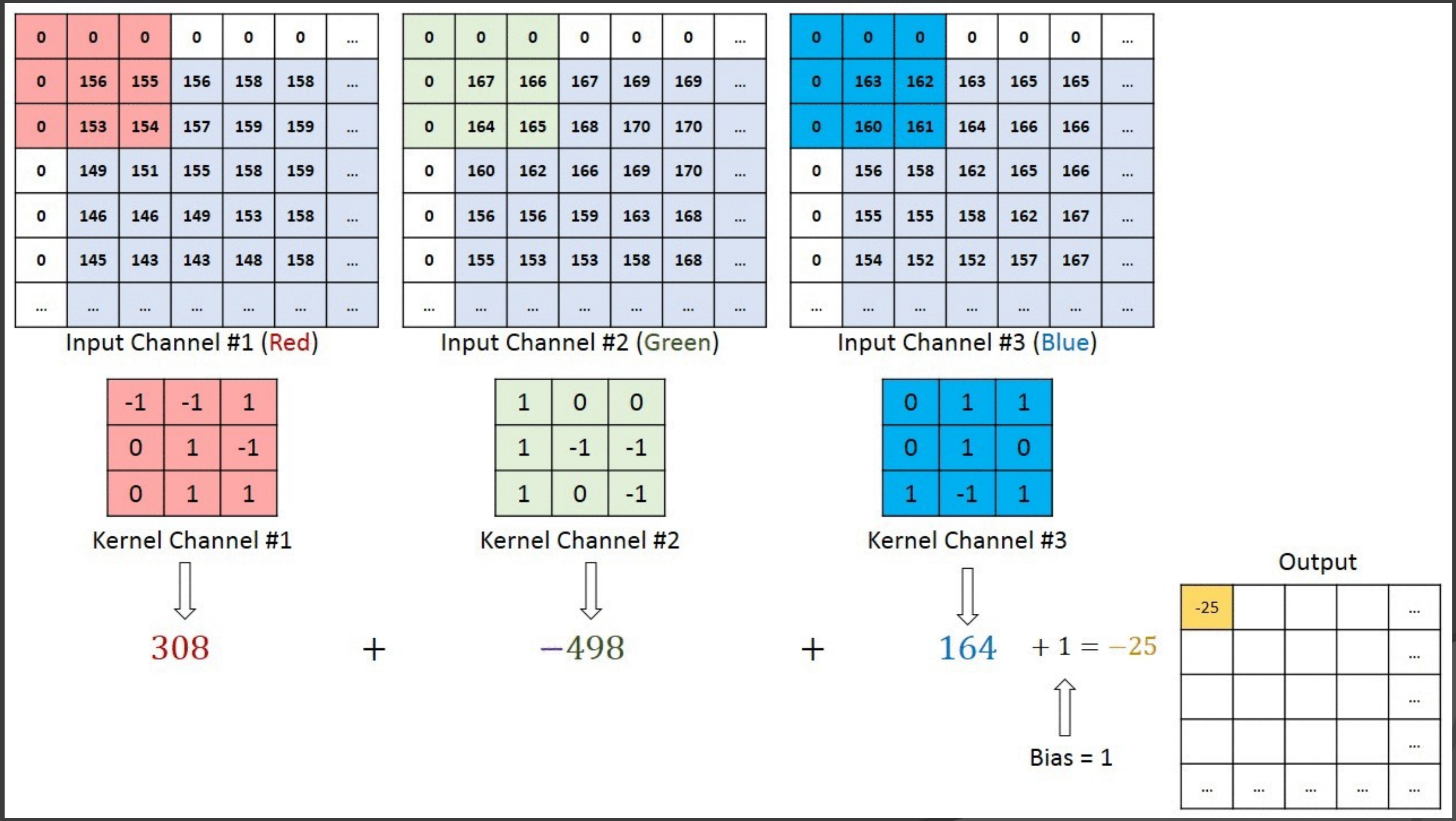


141	157	150	184
58	99	147	181
128	78	92	158

Object Oriented Design Using Android
Leveraging the Strategy Pattern to
Implement Image Filters by Edward C. Epp

Convolution

“Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel”



0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

$$308 + -498 + 164 + 1 = -25$$

Bias = 1

Output

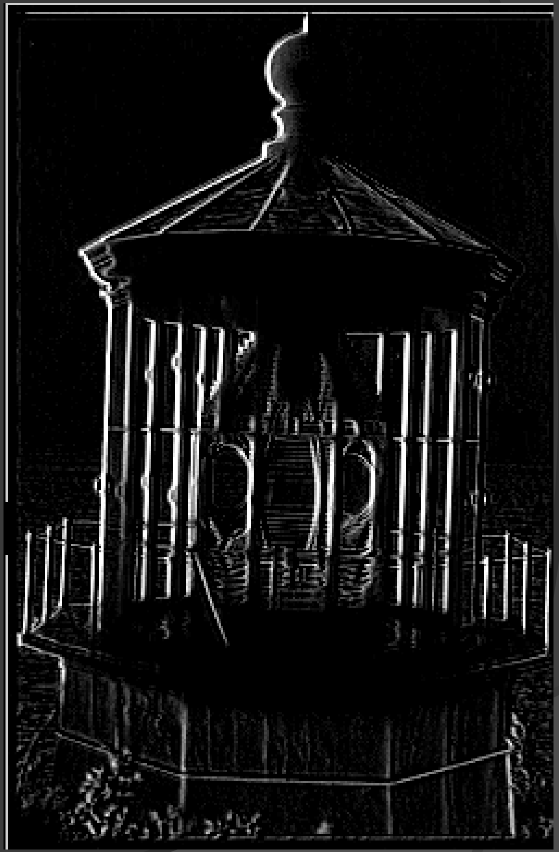
-25				...
				...
				...
				...
...	...	...	...	...

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Edge Detection Convolution

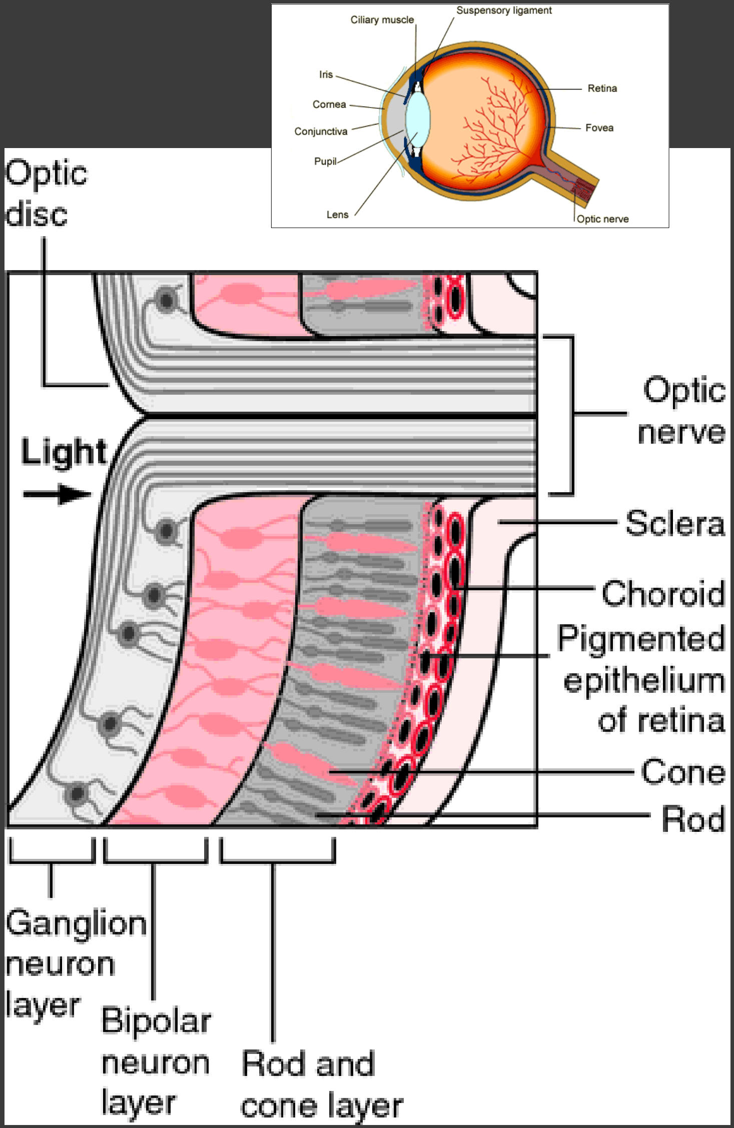
West Edge
Kernel (Filter)

1	1	-1
1	-2	-1
1	1	-1



Object Oriented Design Using Android Leveraging the Strategy Pattern to Implement Image Filters by Edward C. Epp

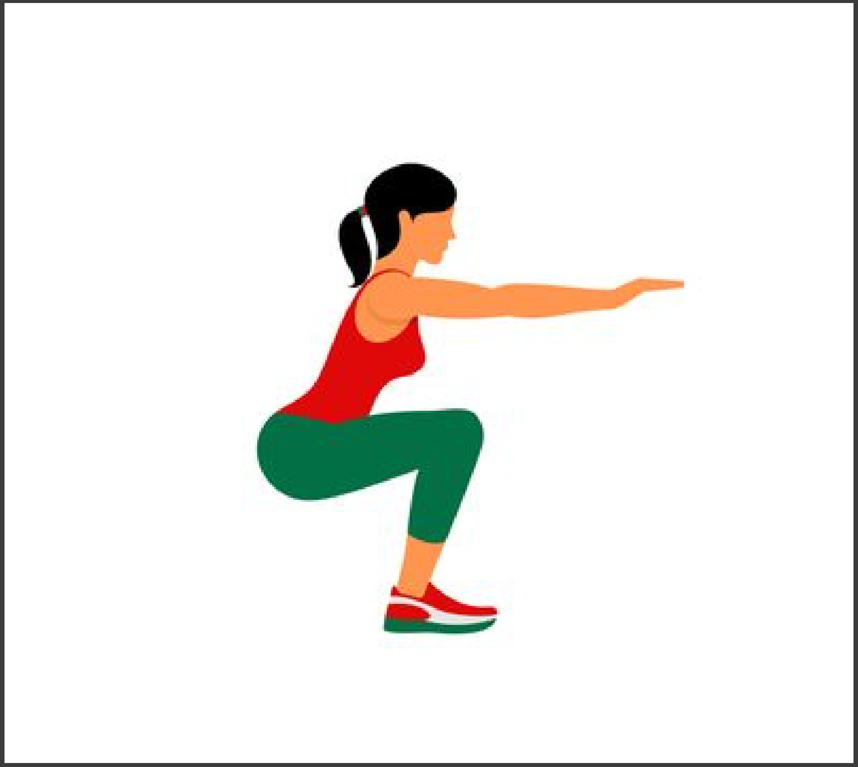
Human Eye



- Three neuron layers in the retina
- Edge, color maps, gradient fills and other convolutions
- Detect movement, shapes, angles
- Auto contrast
- 770,000 to 1,700,000 optic nerve fibers
- 100 times more photoreceptor cells than optic nerve cells (convolution compression)

https://www.researchgate.net/figure/Layers-of-the-Retina_fig4_310101335
https://www.quora.com/What-cells-in-the-eye-are-responsible-for-edge-detection
https://thebrain.mcgill.ca/flash/a/a_02/a_02_cl/a_02_cl_vis/a_02_cl_vis.html
https://en.wikipedia.org/wiki/Optic_nerve
https://en.wikipedia.org/wiki/Retina
https://socratic.org/questions/what-are-the-functions-of-the-cornea-pupil-lens-retina-and-optic-nerve

Exercises 2



https://coach.nine.com.au/fitness/10-exercises-for-a-full-body-workout-at-home/7fc517cc-3e33-4b2d-ab9f-7fe989f3e5c9

Convolution

1. Draw the image on the right as it would look if it where printed. Use 10 for the value of white and 0 for black
2. Do a convolution of the image by applying the kernel on the left to the gray pixels in the image on the right to create a new image. Enter each derived pixel into an appropriate square in a 6x5 matrix
3. What will the resulting image look like? Draw it. Use 10 or greater as the value for white and 0 or less for black
4. Start with an image that is completely white (i.e., all of its pixels have a value of 10). Do the convolution below.
5. What will happen if all the pixels in the image are black (0)?
6. Compare the answers in 4 and 5 and explain why the results are what they age

Image

0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10
0	0	0	0	10	10	10	10

Kernel

-1	-1	-1
-1	8	-1
-1	-1	-1

https://en.wikipedia.org/wiki/Kernel_(image_processing)

Convolution cont.

What do you think the following kernel’s do?

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

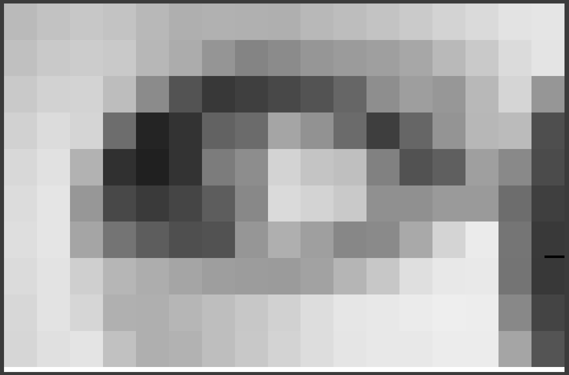
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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

https://en.wikipedia.org/wiki/Kernel_(image_processing)

A Single Artificial Neuron

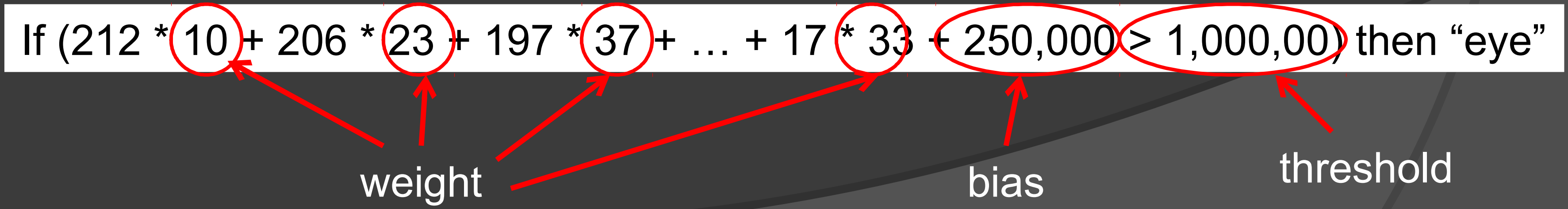
Contrived example



Transform 2D image to one dimension

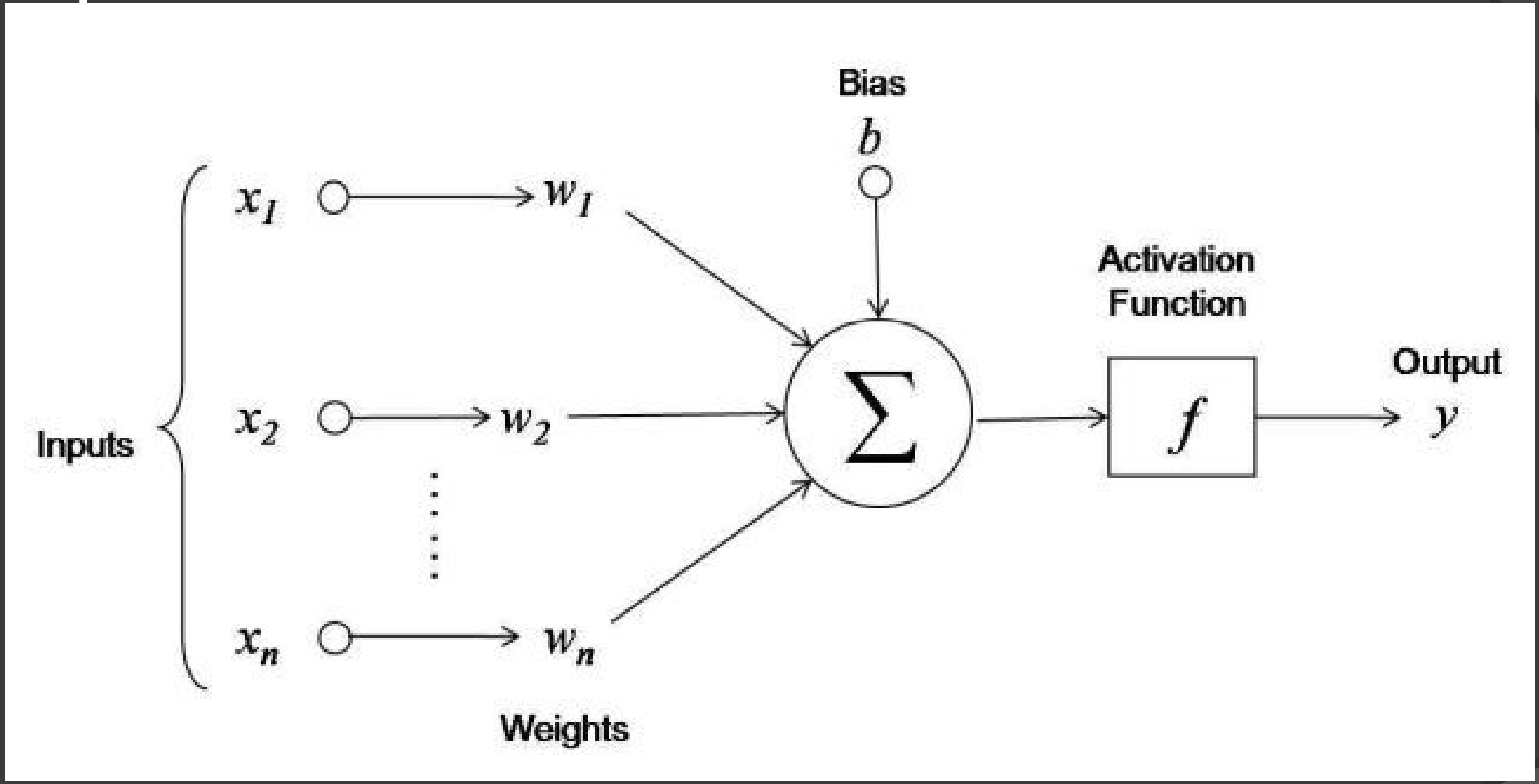
212	206	197	…	17

Inference by one of many output layer neurons



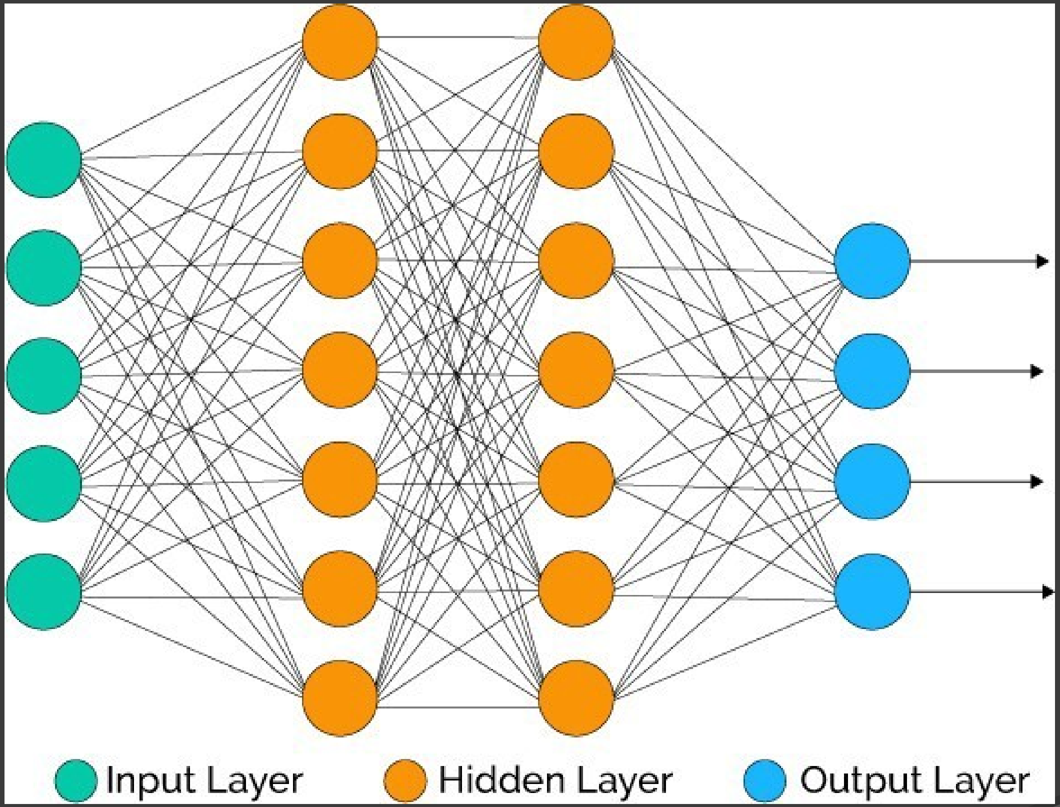
If (212 * 10 + 206 * 23 + 197 * 37 + … + 17 * 33 + 250,000 > 1,000,00) then “eye”

Formal Artificial Neuron Representation



https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf

Artificial Neural Network

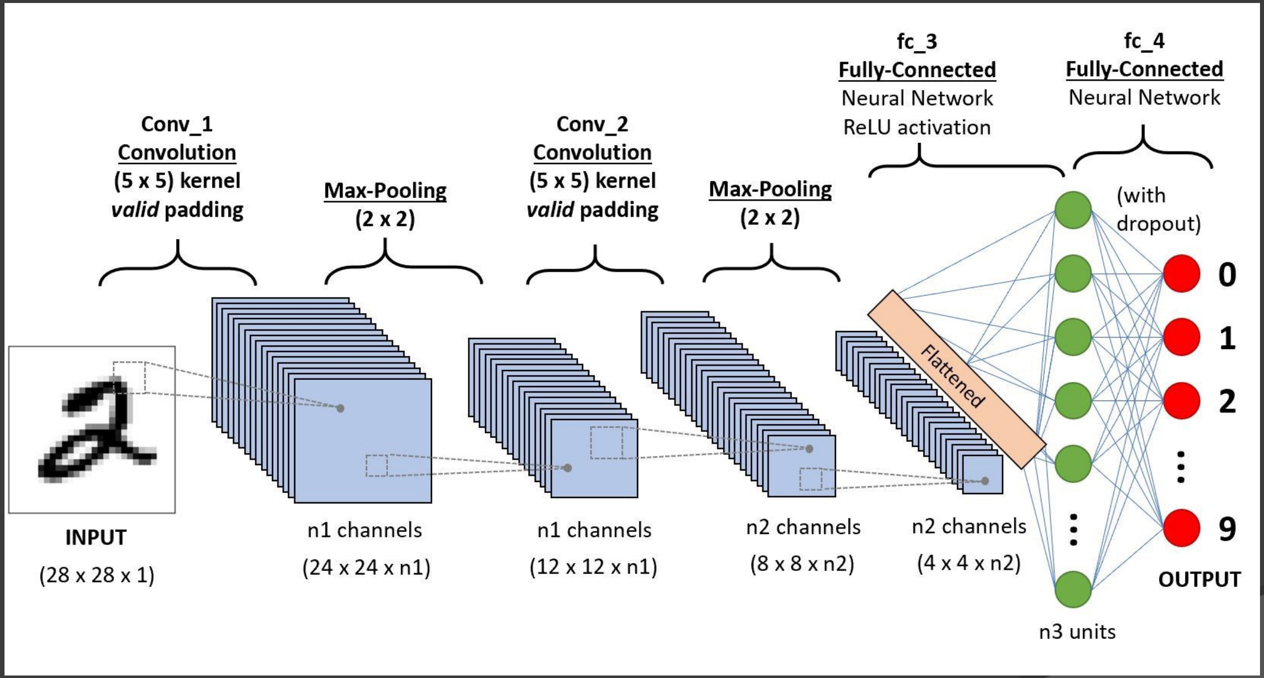


https://medium.com/datadriveninvestor/when-not-to-use-neural-networks-89fb50622429

Convolutional Neural Network CNN

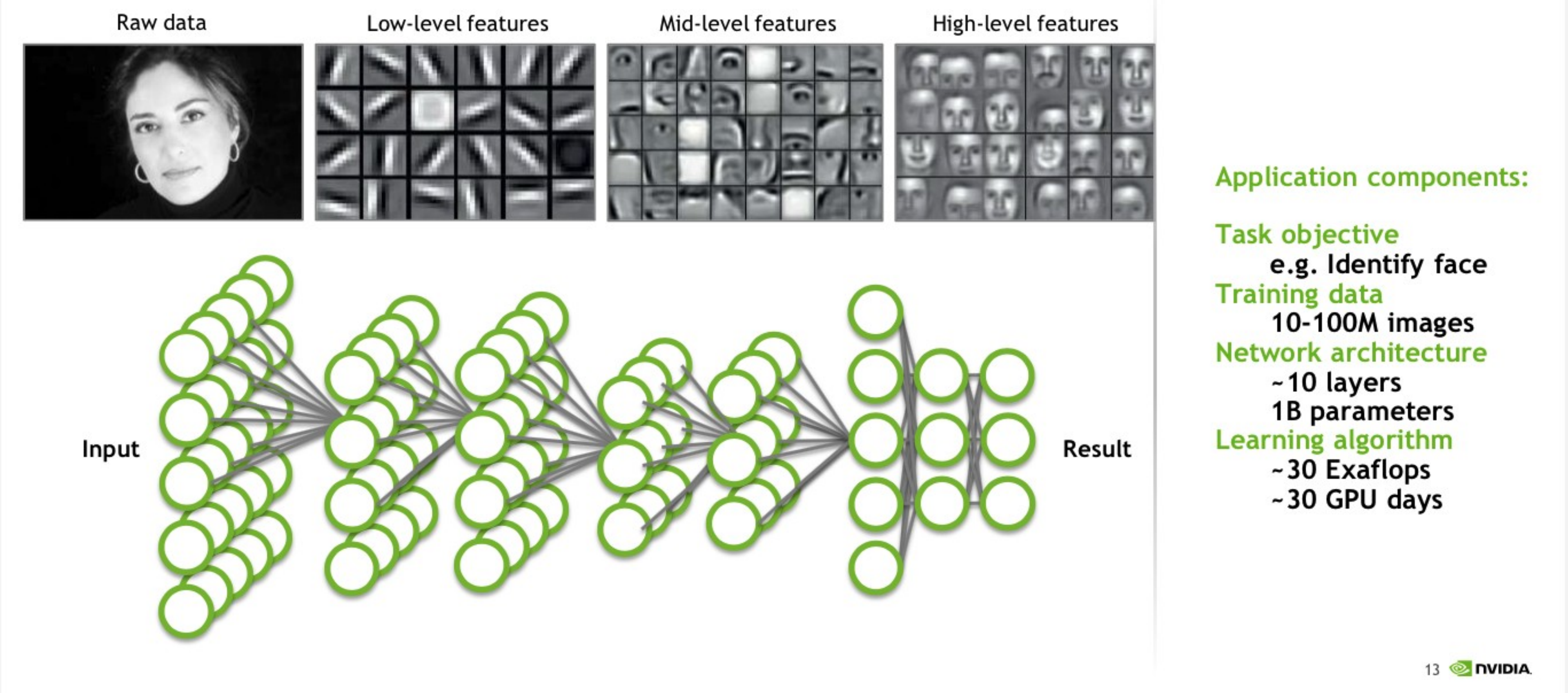
Manage data bandwidth

- Extracting features
- Subsampling images
- Preprocess data at the sensor



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Deep Neural Network (DNN)



Application components:

Task objective
e.g. Identify face

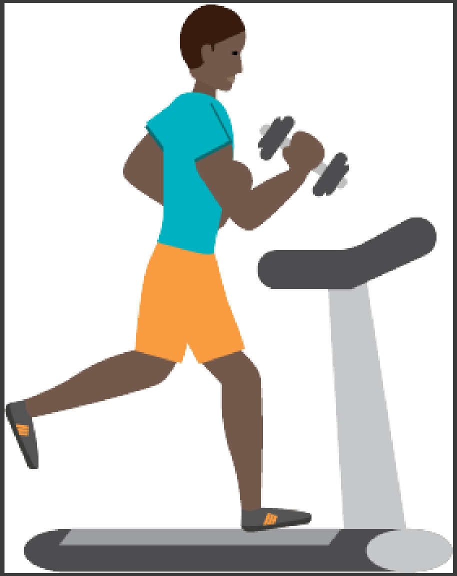
Training data
10-100M images

Network architecture
~10 layers
1B parameters

Learning algorithm
~30 Exaflops
~30 GPU days

https://app.box.com/s/r159kx1ap90ursq16uad70hl3054es34

Exercises 3

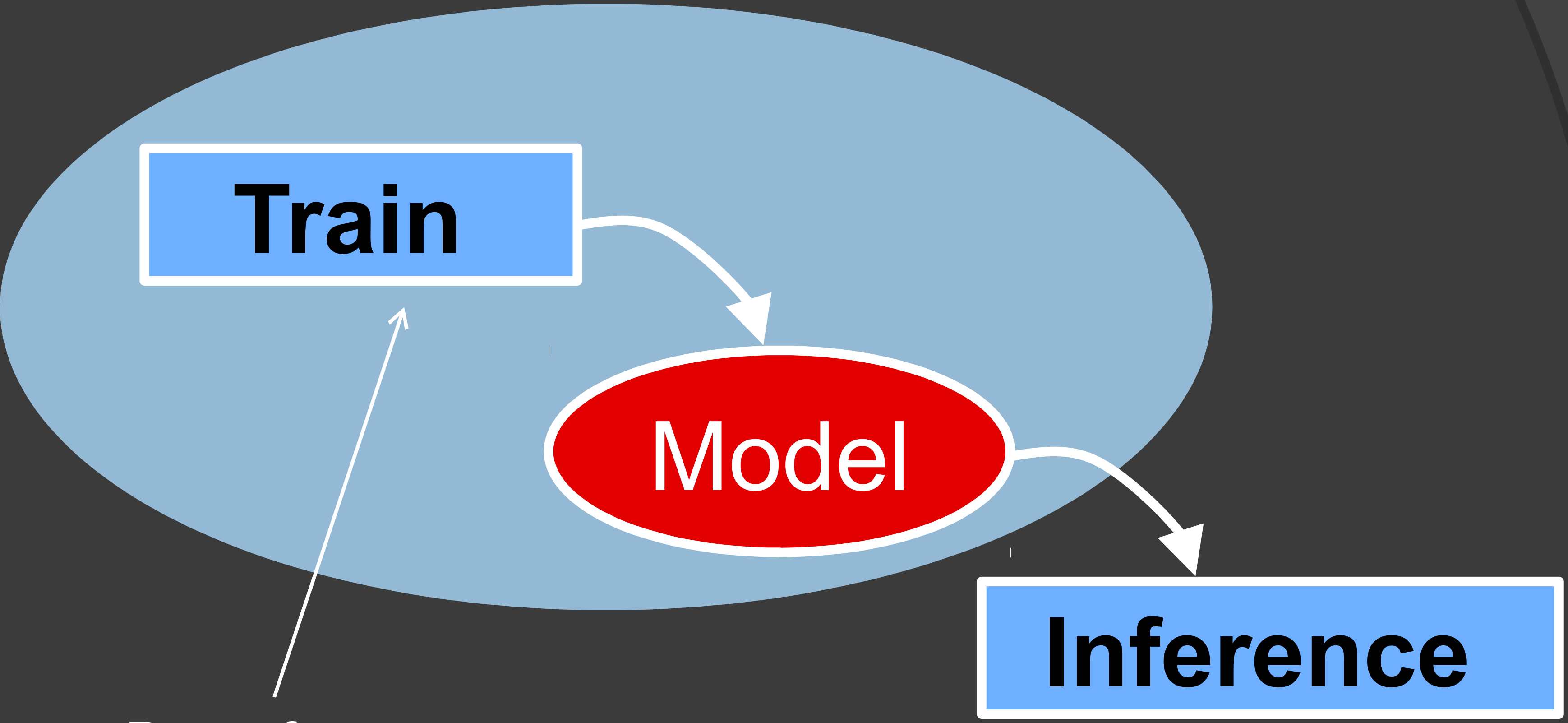


https://freepngimg.com/png/52373-exercise-image-free-download-png-hd

Neural Networks TBD

1. Question

Training



Done for us

Training Process

- Collect dozens to millions of images
- Partition them into training and test sets
- Start with dozens or hundreds of convolution kernels
- Start with hundred or thousands of neurons
- Repeat – each repetition is called an epoch
  - Use the training set to made small adjustments to the convolution kernels and neuron weights and biases
  - Use the test set to evaluate the inferences
  - Leave loop when the training criteria are met
- Save the results as a model

Evaluating Deep Learn Decisions

How do you reverse engineer a decision so you can justify, defend, debug, test it?

How does that make you feel?

Do Convolution Neural Networks meet your definition of intelligence? Why”

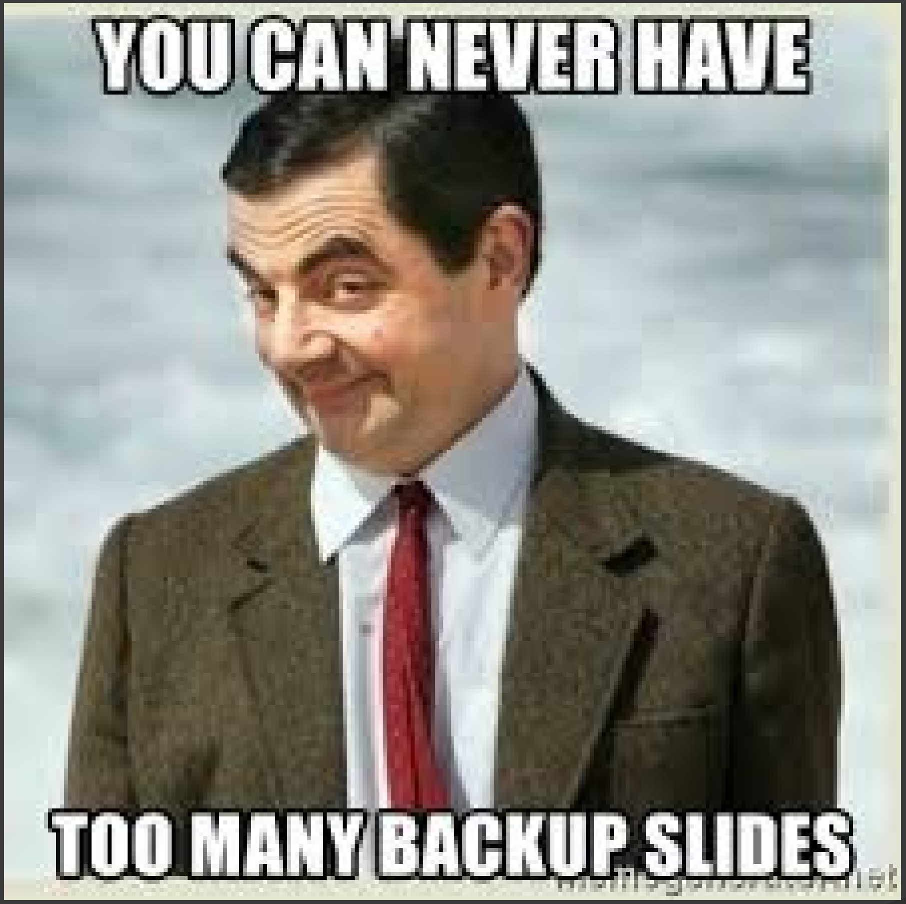
Brains vs. Computer

Not a good comparison

- Brains are analogue: computers are digital
- The brains uses content-addressable memory
- The brain massively parallel
- Synapses are far more complex than electrical logic gates
- Brains have bodies

https://scienceblogs.com/developingintelligence/2007/03/27/why-the-brain-is-not-like-a-co

Backup Slides



https://memegenerator.net/instance/74725371/mr-bean-you-can-never-have-too-many-backup-slides

References

http://blog.ruofeidu.com/quick-tutorial-on-refereces-management-using-mendeley-endnote-word-sharelatex/

Udacity Tutorials

- Introduction to TensorFlow Lite
  - https://www.udacity.com/course/intro-to-tensorflow-lite--ud190
  - By TensorFlow Lite

- Intro to TensorFlow for Deep Learning
  - By TensorFlow
  - https://www.udacity.com/course/intro-to-tensorflow-for-deep-learning--ud187

API resources

- https://github.com/google/ftc-object-detection/tree/master/TFObjectDetector
- http://ftckey.com/apis/ftc/
- https://github.com/ftctechnh/ftc_app/wiki/Using-TensorFlow-Lite-for-Mineral-Detection
- https://github.com/google/ftc-object-detection
- https://github.com/google/ftc-object-detection/tree/master/TFObjectDetector
- https://github.com/ftctechnh/ftc_app
- https://github.com/ftctechnh/ftc_app/tree/master/FtcRobotController/src/main/assets
- https://github.com/ftctechnh/ftc_app/wiki/Java-Sample-TensorFlow-Object-Detection-Op-Mode
- https://github.com/ftctechnh/ftc_app/blob/master/FtcRobotController/src/main/java/org/firstinspires/ftc/robotcontroller/external/samples/ConceptTensorFlowObjectDetection.java

TensorFlow

- https://github.com/ftctechnh/ftc_app/wiki/Java-Sample-TensorFlow-Object-Detection-Op-Mode
- https://www.firstinspires.org/resource-library/ftc/technology-information-and-resources
- https://github.com/ftctechnh/ftc_app/blob/master/FtcRobotController/src/main/java/org/firstinspires/ftc/robotcontroller/external/samples/ConceptTensorFlowObjectDetection.java
- https://www.tensorflow.org/lite
- https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2-tflite/#0
- https://www.tensorflow.org/lite/overview
- https://medium.com/@rdeep/tensorflow-lite-tutorial-easy-implementation-in-android-145443ec3775

OnBot Java Brief Reference

- https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/ftc-onbot-java-tutorial.pdf

SKYSTONE℠

PRESENTED BY Qualcomm

FIRST® RISE℠ powered by *Star Wars: Force for Change*

2019-2020 *FIRST*® Tech Challenge

OnBot Java Tutorial

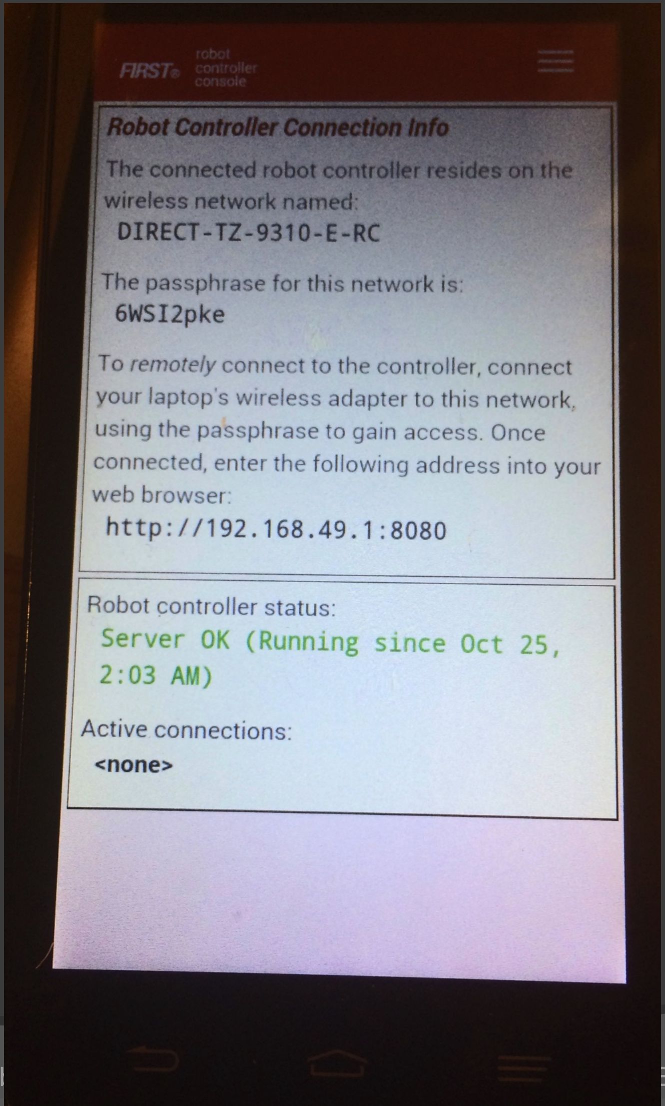
Start Up the Robot Controller

- Boot Robot Controller
- Start up FTC Robot Controller app

Start up the Driver Station

- Boot Driver Station
- Start up FTC Driver Station app
- Go to settings
  - Pair with Robot Controller
    - Turn Filter off
      - Select your robot controller

Driver Station



Robot Controller Connection Info

The connected robot controller resides on the wireless network named:

DIRECT-TZ-9310-E-RC

The passphrase for this network is:

6WSI2pke

To *remotely* connect to the controller, connect your laptop's wireless adapter to this network, using the passphrase to gain access. Once connected, enter the following address into your web browser:

http://192.168.49.1:8080

Robot controller status:

Server OK (Running since Oct 25, 2:03 AM)

Active connections:

<none>

Bring up the Program & Manage window

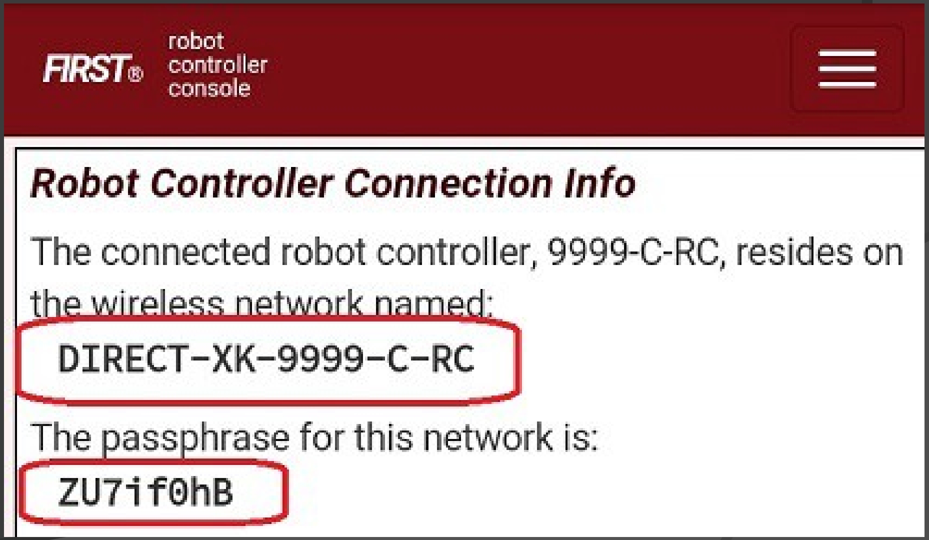
Note

- Your Network name
- Network Password
- Local network server URL

On Your Laptop

- Disconnect from any global networks
- Connect to your Direct WiFi network
- See the Program & Manage menu on the Driver Station if you need a password
- Connect to this network

Page 104



Robot Controller Connection Info

The connected robot controller, 9999-C-RC, resides on the wireless network named:

DIRECT-XK-9999-C-RC

The passphrase for this network is:

ZU7if0hB

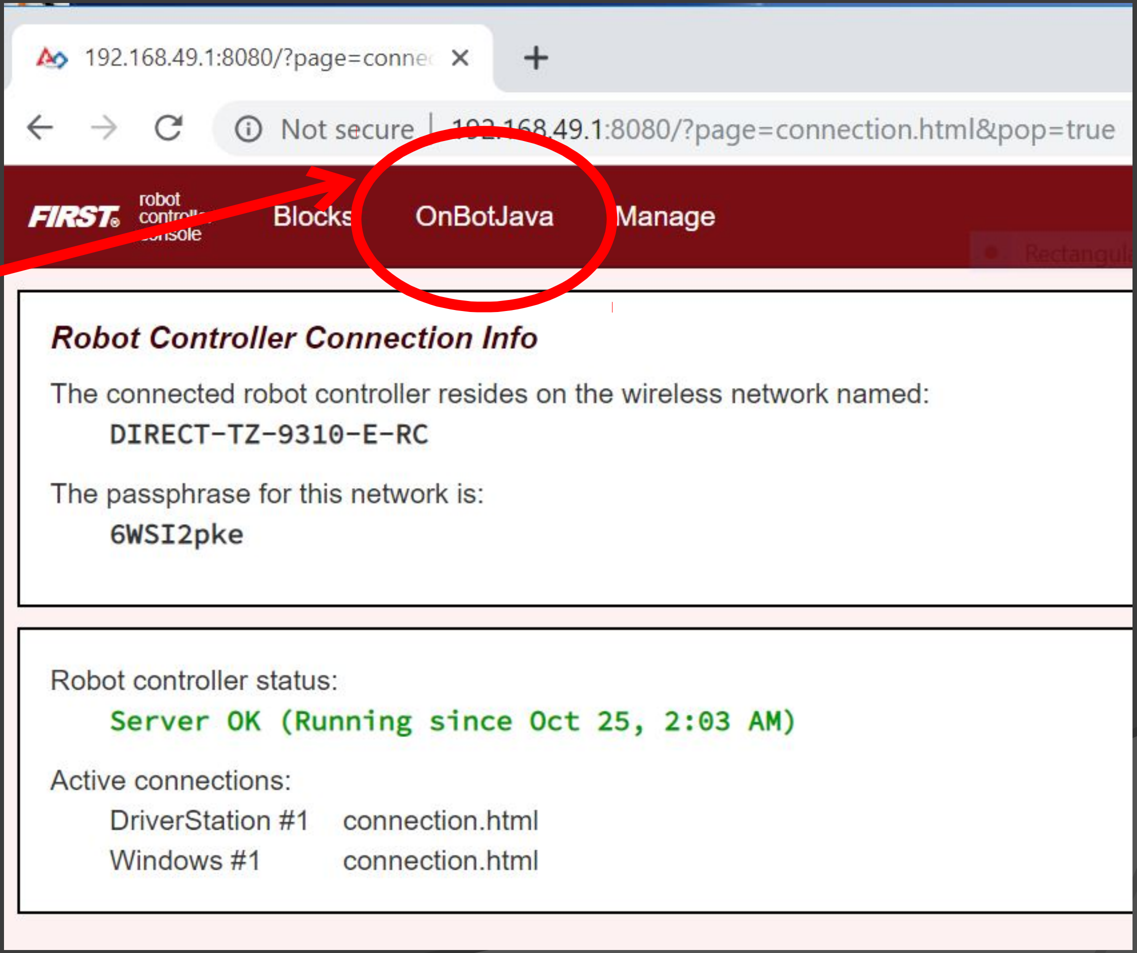
Starting Up OnBot Java

Page 110 & 111

- In Chrome on your laptop
- http://192.168.49.1:8080

Your are Connected to OnBot Java Server

Select OnBot Java Programming mode



192.168.49.1:8080/?page=conne
Not secure | 192.168.49.1:8080/?page=connection.html&pop=true
FIRST robot controller console
Blocks
OnBotJava
Manage

Robot Controller Connection Info

The connected robot controller resides on the wireless network named:

```
DIRECT-TZ-9310-E-RC
```

The passphrase for this network is:

```
6WSI2pke
```

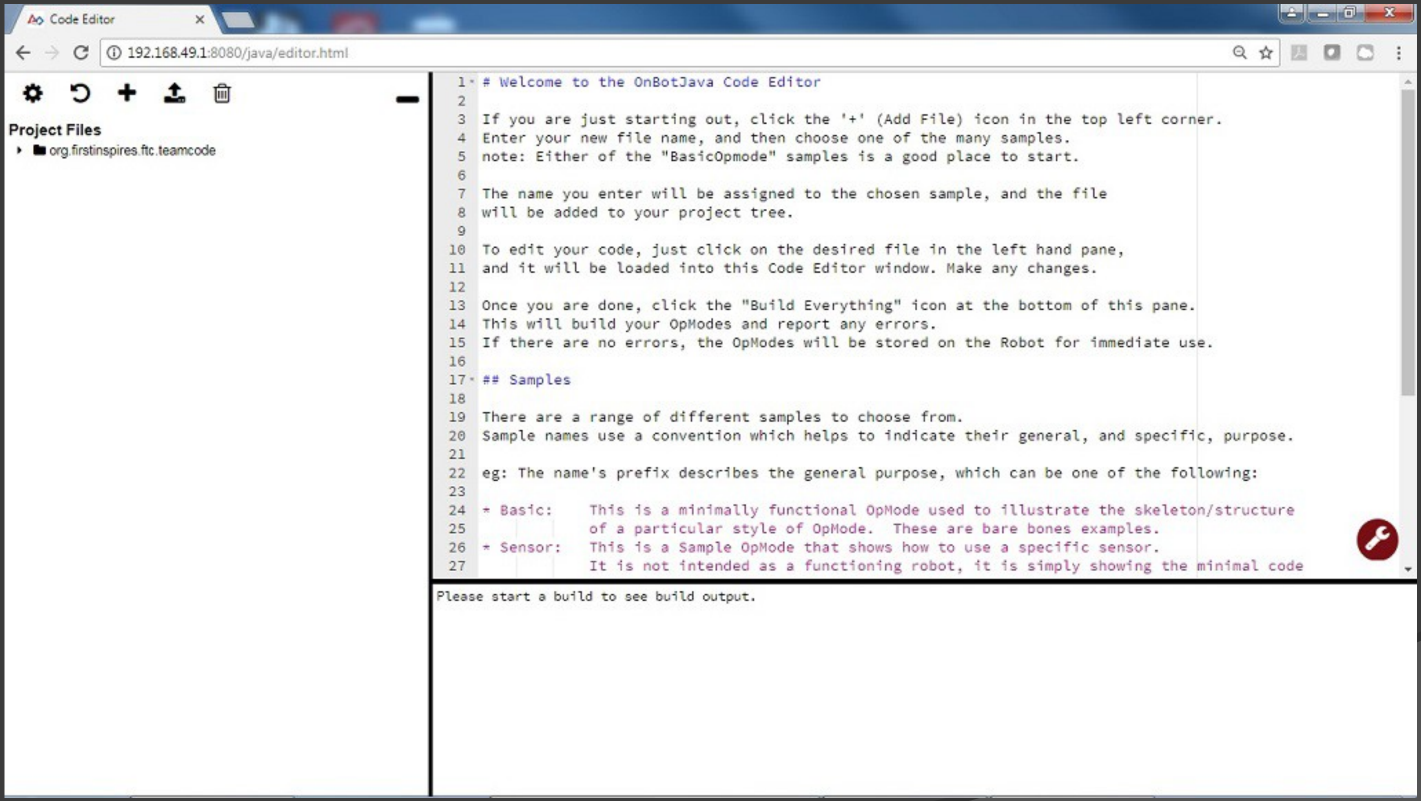
Robot controller status:

```
Server OK (Running since Oct 25, 2:03 AM)
```

Active connections:

DriverStation #1	connection.html
Windows #1	connection.html

Starting Screen

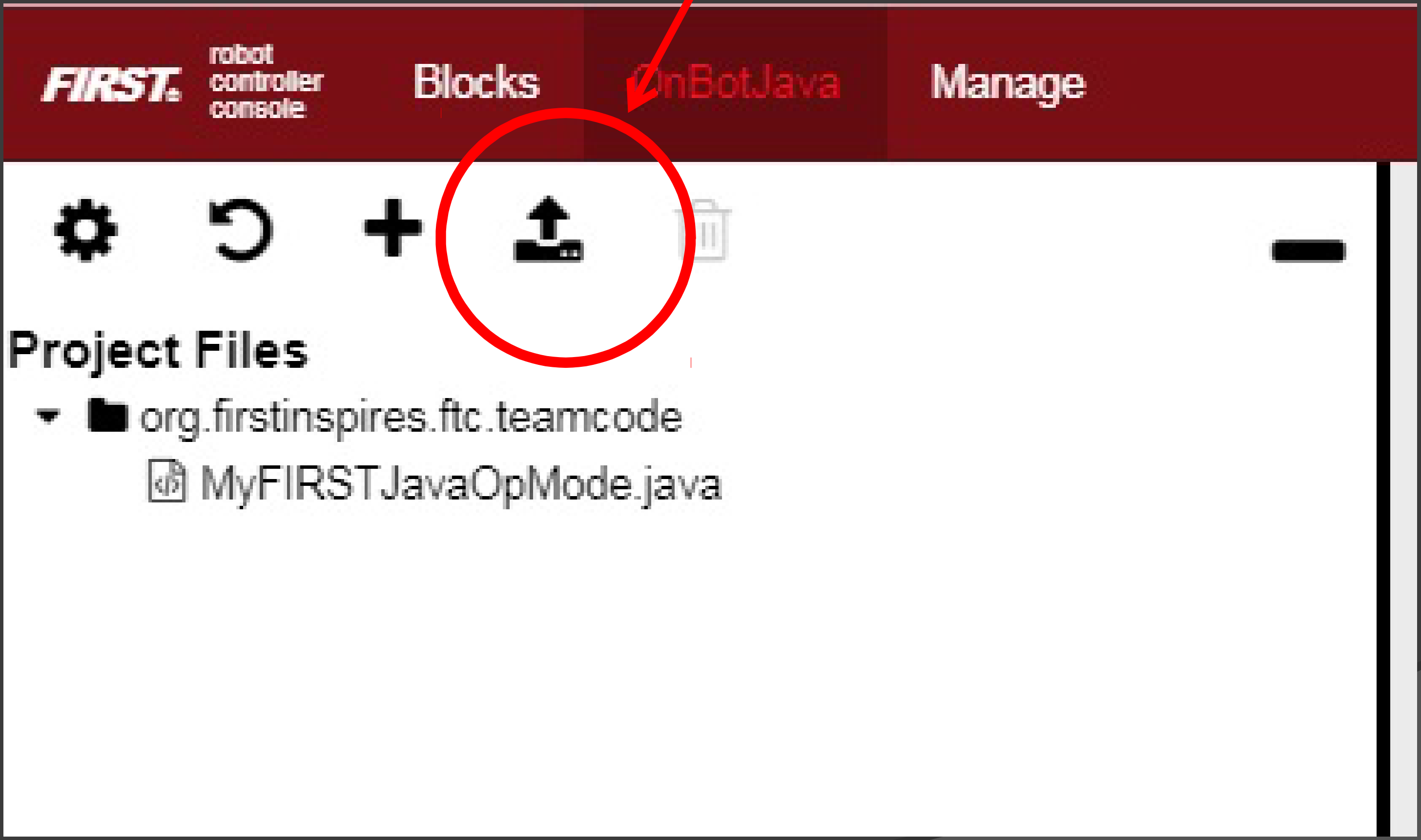

Code Editor
192.168.49.1:8080/java/editor.html
Project Files
org.firstinspires.ftc.teamcode

```
# Welcome to the OnBotJava Code Editor

If you are just starting out, click the '+' (Add File) icon in the top left corner.
Enter your new file name, and then choose one of the many samples.
note: Either of the "BasicOpmode" samples is a good place to start.

The name you enter will be assigned to the chosen sample, and the file
will be added to your project tree.

To edit your code, just click on the desired file in the left hand pane,
and it will be loaded into this Code Editor window. Make any changes.

Once you are done, click the "Build Everything" icon at the bottom of this pane.
This will build your OpModes and report any errors.
If there are no errors, the OpModes will be stored on the Robot for immediate use.

## Samples

There are a range of different samples to choose from.
Sample names use a convention which helps to indicate their general, and specific, purpose.

eg: The name's prefix describes the general purpose, which can be one of the following:

* Basic:    This is a minimally functional OpMode used to illustrate the skeleton/structure
            of a particular style of OpMode.  These are bare bones examples.
* Sensor:   This is a Sample OpMode that shows how to use a specific sensor.
            It is not intended as a functioning robot, it is simply showing the minimal code
```

```
Please start a build to see build output.
```

Dashboard

Upload Java File

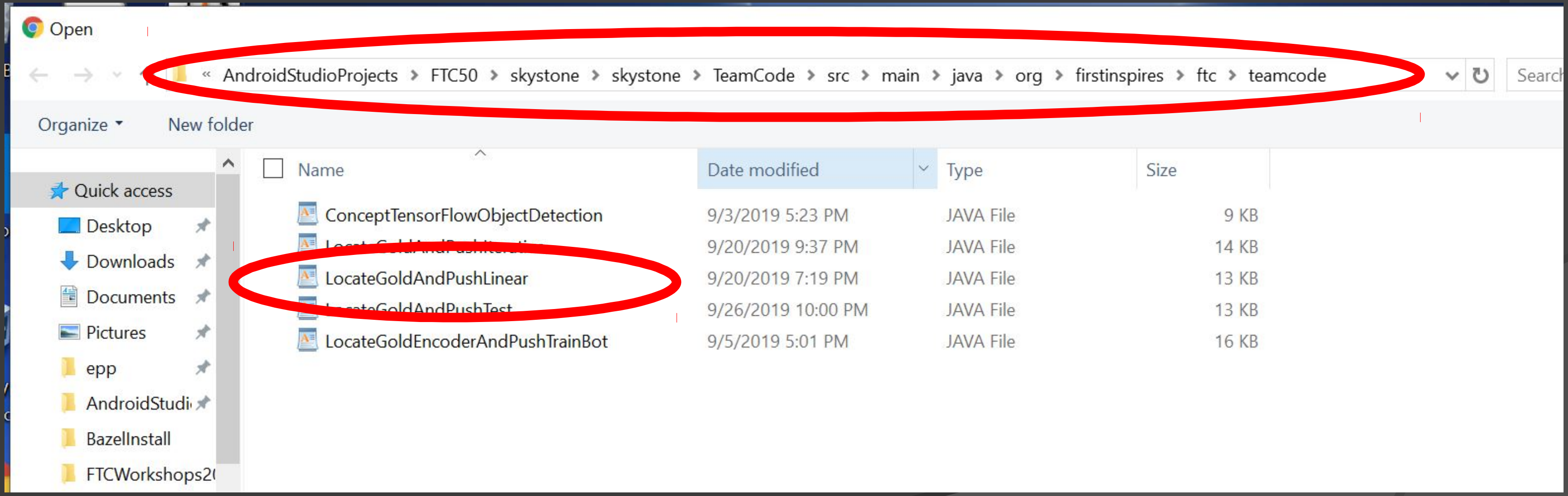


Project Files

- org.firstinspires.ftc.teamcode
  - MyFIRSTJavaOpMode.java

Open FTC Java File

LocateGoldAndPushLinear.java



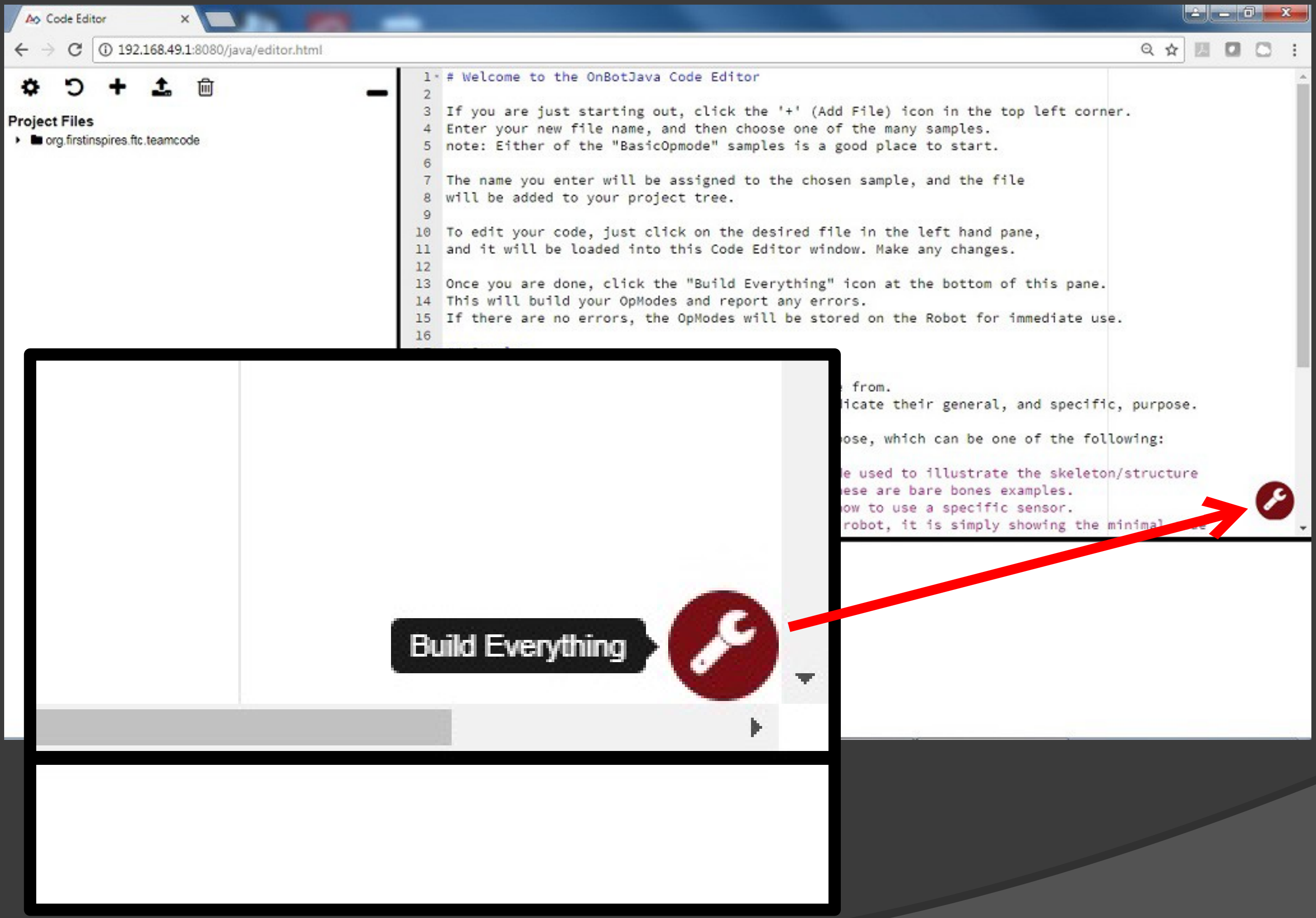
Open

« AndroidStudioProjects › FTC50 › skystone › skystone › TeamCode › src › main › java › org › firstinspires › ftc › teamcode

Name	Date modified	Type	Size
ConceptTensorFlowObjectDetection	9/3/2019 5:23 PM	JAVA File	9 KB
LocateGoldAndPushIteration	9/20/2019 9:37 PM	JAVA File	14 KB
LocateGoldAndPushLinear	9/20/2019 7:19 PM	JAVA File	13 KB
LocateGoldAndPushTest	9/26/2019 10:00 PM	JAVA File	13 KB
LocateGoldEncoderAndPushTrainBot	9/5/2019 5:01 PM	JAVA File	16 KB

Build

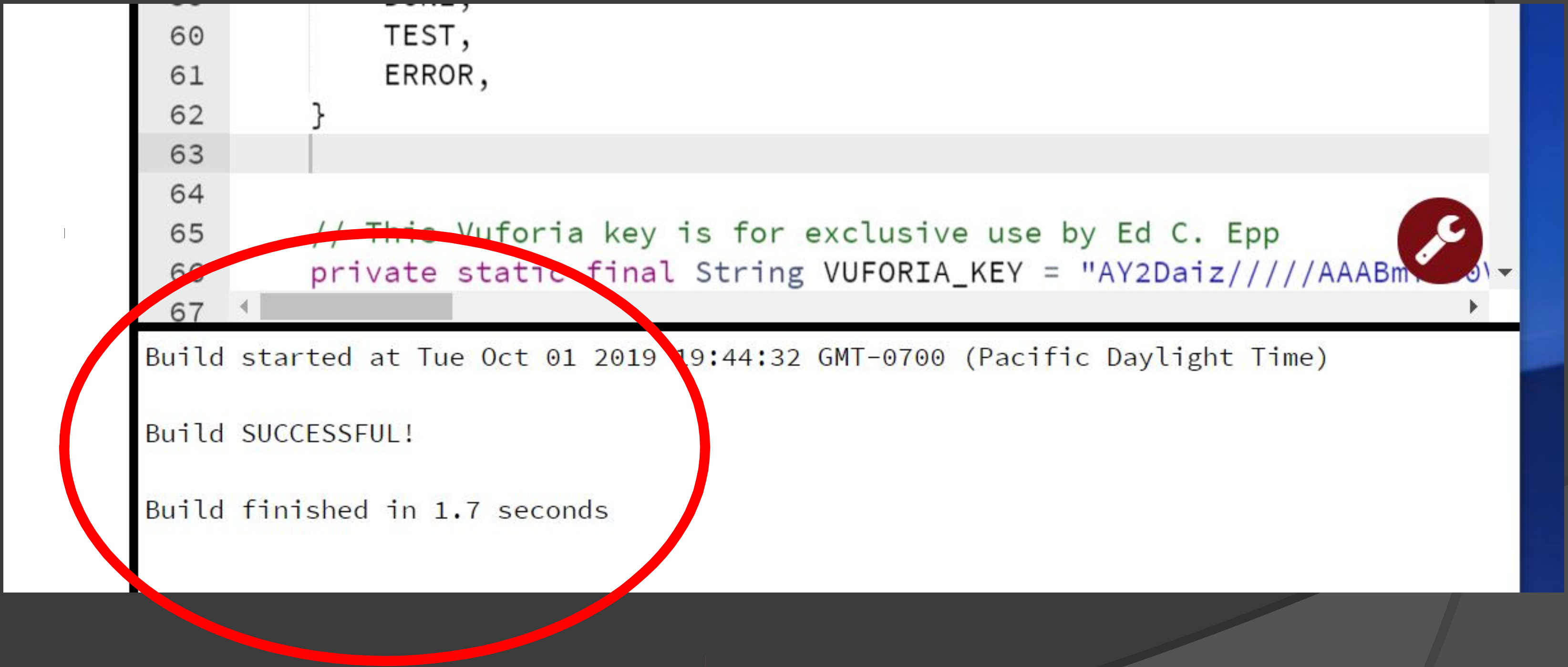
Page 121



Code Editor
192.168.49.1:8080/java/editor.html
Project Files
org.firstinspires.ftc.teamcode
Build Everything

```
# Welcome to the OnBotJava Code Editor

If you are just starting out, click the '+' (Add File) icon in the top left corner.
Enter your new file name, and then choose one of the many samples.
note: Either of the "BasicOpmode" samples is a good place to start.

The name you enter will be assigned to the chosen sample, and the file
will be added to your project tree.

To edit your code, just click on the desired file in the left hand pane,
and it will be loaded into this Code Editor window. Make any changes.

Once you are done, click the "Build Everything" icon at the bottom of this pane.
This will build your OpModes and report any errors.
If there are no errors, the OpModes will be stored on the Robot for immediate use.

                                                  e from.
                                                  dicate their general, and specific, purpose.

                                                  pose, which can be one of the following:

                                                  de used to illustrate the skeleton/structure
                                                  hese are bare bones examples.
                                                  how to use a specific sensor.
                                                   robot, it is simply showing the minimal
```

Build Success



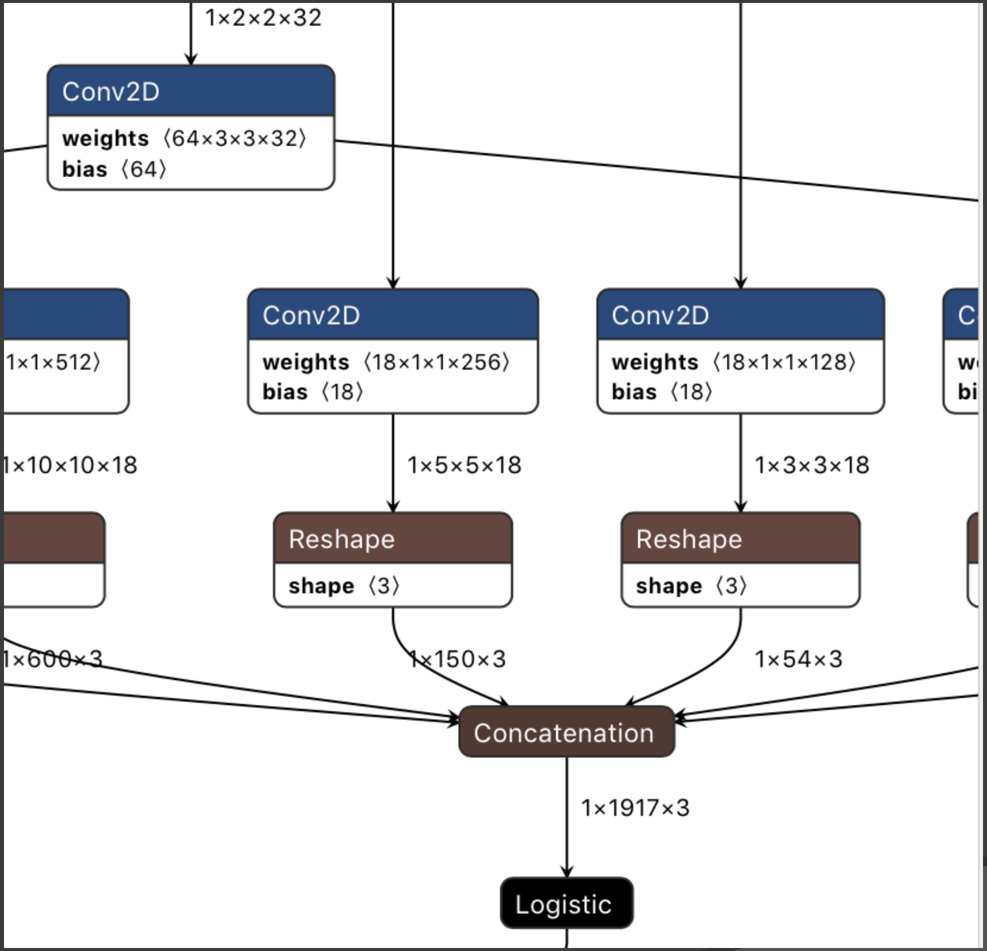
```
        TEST,
        ERROR,
    }


    // This Vuforia key is for exclusive use by Ed C. Epp
    private static final String VUFORIA_KEY = "AY2Daiz////AAABm
```

```
Build started at Tue Oct 01 2019 19:44:32 GMT-0700 (Pacific Daylight Time)

Build SUCCESSFUL!

Build finished in 1.7 seconds
```

OnBot Java Tutorials

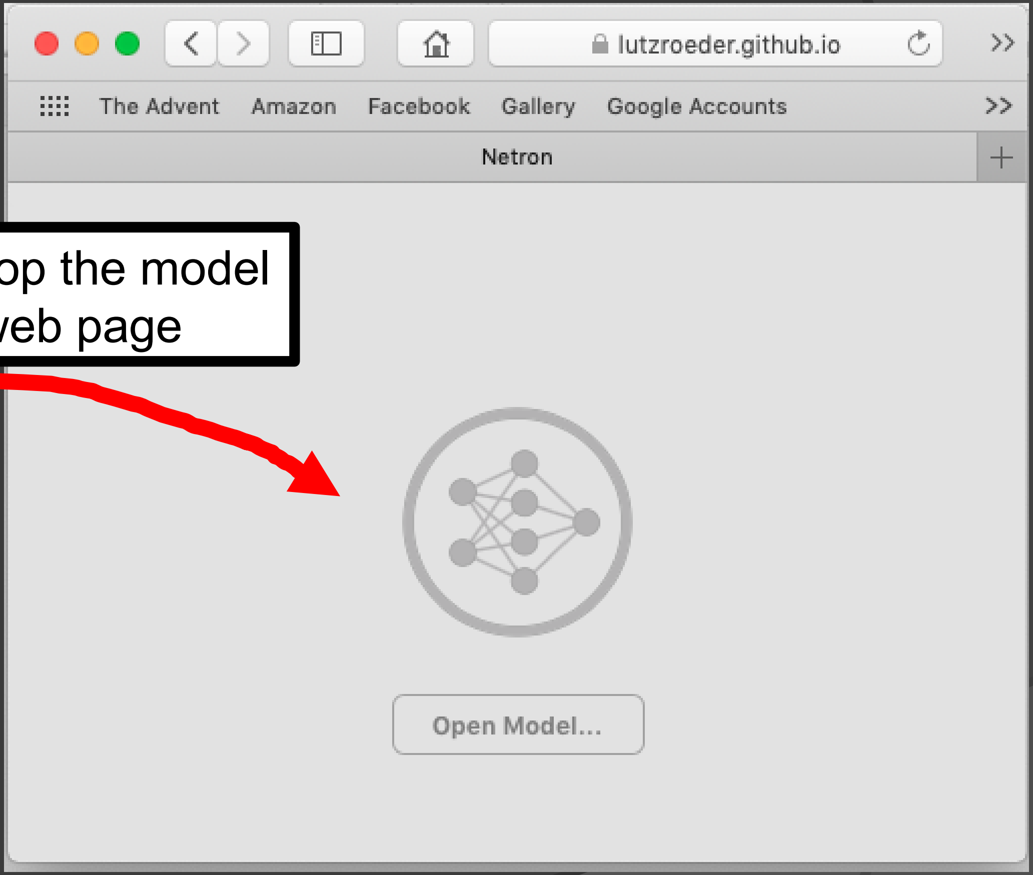
- Basic OnBot Java tutorial.
  - https://github.com/ftctechnh/ftc_app/wiki/OnBot-Java-Tutorial
- This one is much more complete at a 140 pages
  - https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/ftc-onbot-java-tutorial.pdf

Netron Model Visualizer



Netron Browser Version

https://lutzroeder.github.io/netron/

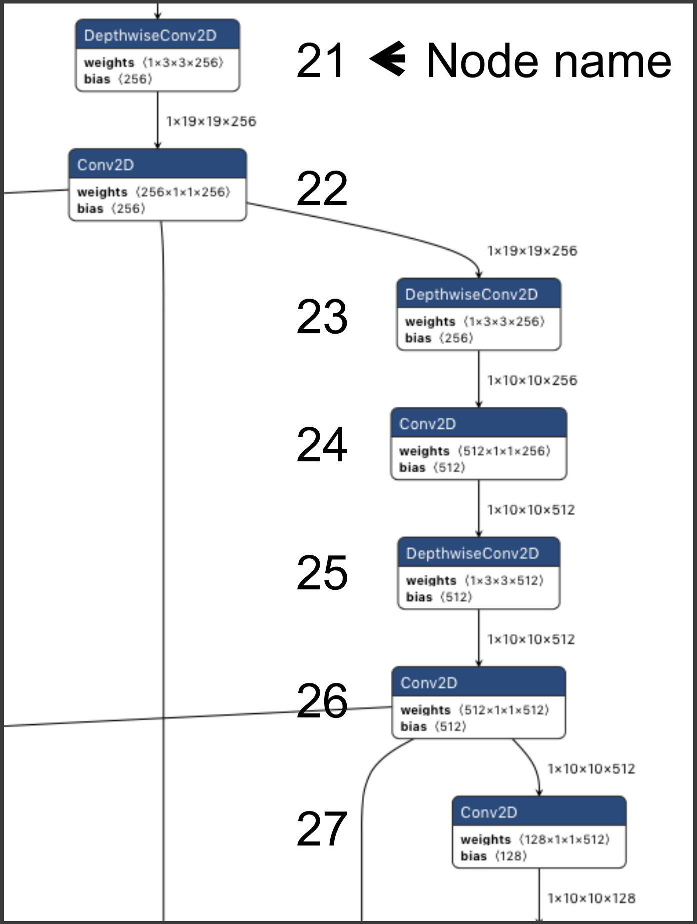


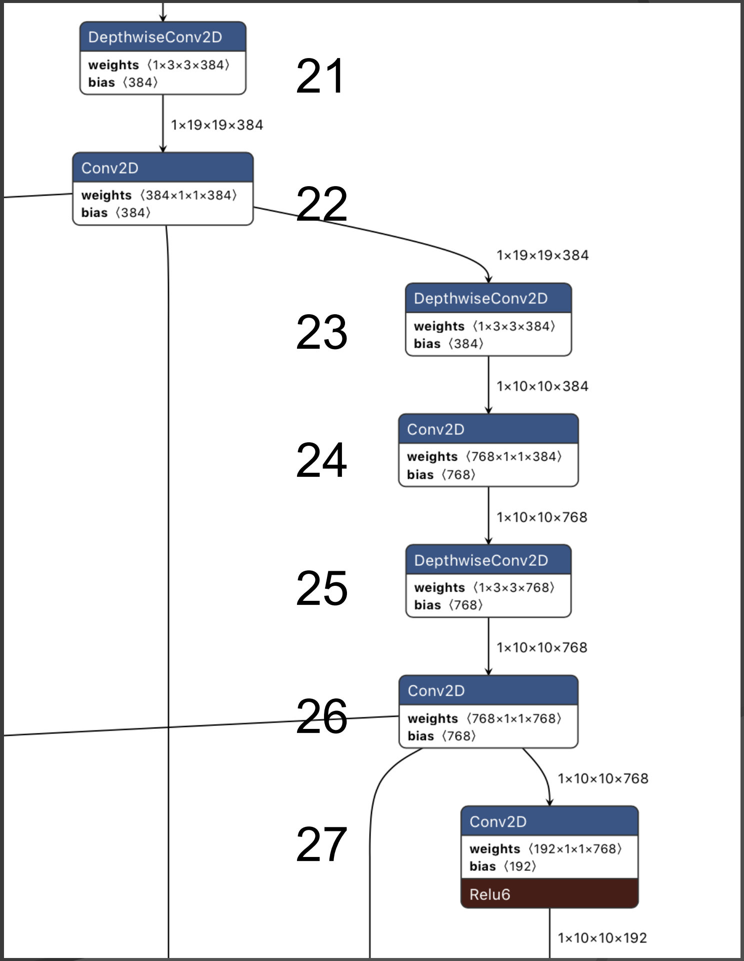
Drag and drop the model on the web page

Model file

Skystone.tflite

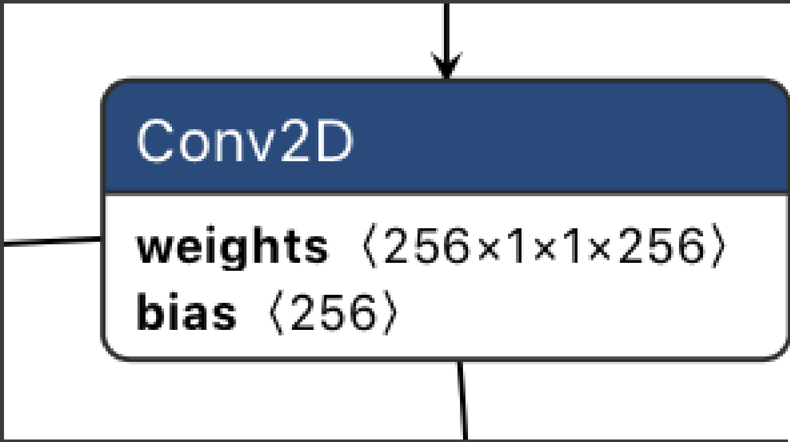
RoverRuckus and Skystone Models Structure is the Same





Learning by Adjusting Weights and Bias's

Weights and bias's are different for the two models



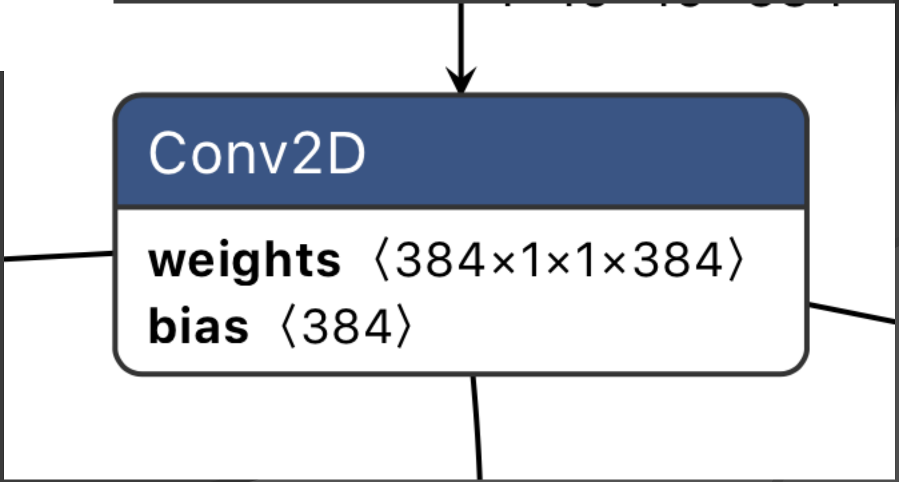
Conv2D

weights ⟨256×1×1×256⟩

bias ⟨256⟩

RoverRuckus

Skystone



Conv2D

weights ⟨384×1×1×384⟩

bias ⟨384⟩

Node 22

Convolution Answers

An edge detected

Question 2

0	0	-30	30	0	0
0	0	-30	30	0	0
0	0	-30	30	0	0
0	0	-30	30	0	0
0	0	-30	30	0	0

Question 4

The transformed pixels are all black (0).

Question 5

The transformed pixels are all black (0).

Question 6

The images are uniform. There are no edges.

Exercise 5

- Enhance the targetGold turn

Recognition: estimateAngleToObject , getBotton, getConfidence, getImageHeight, getImageWidth, getLable, getLeft, getRight, getTop, getWidth

REV 41 1300 Core Hex Motor: 288 counts / rotation

REV 41 1354 90 mm dia Traction Wheel

pi d = 282.7 mm / rotation

282.7 mm / rotation / 288 counts / rotation = 0.9817 mm / count

Or about 1 mm / count

Distance between wheels = 8 ¼ inches = 209.55 mm

pi d = pi 209.55 mm = 658.32 mm / rotation

658.32 mm / rotation / 0.9817 mm / count = 670.6 counts / rotation

Where do your attribute error (the omni wheels)

Background

Disclaimer

Do not distribute these slides. I borrowed many illustrations from the net. They are all credited but I am using them without permission. This slide set needs to be sanitized.

I plan to put this under a Creative Commons license of some sort once I resolve the illustration issues.

cc Edward C. Epp, 2019

Fastest-Growing Jobs of 2018

- Machine Learning Engineer (12X growth)
  - Top Skills: Deep Learning, Machine Learning, Tensorflow, Apache Spark, Natural Language Processing
  - Where They Work: Apple, Intel, NVIDIA
  - Top Industries: Computer Software, Internet, Information Technology & Services
  - Cities Where Demand is High: San Francisco, Denver, Austin

- Machine Learning Specialist (6X growth)
  - Top Skills: Machine Learning, Deep Learning, Tensorflow, Python, Artificial Intelligence
  - Where They Work: Google, Amazon, Apple
  - Top Industries: Computer Software, Higher Education, Internet
  - Cities Where Demand is High: San Francisco, New York City, Madison

https://www.techrepublic.com/article/ai-skills-reign-supreme-in-the-fastest-growing-jobs-of-the-year/

What you will learn

- Read code – know what details to ignore for now
- Don’t assume intelligence because it’s a computer or says smart
- Something's are surprisingly the same and other thing surprisingly different