MACHINE LEARNING FOR ANDROID APPLICATIONS

Instructor: Christos Kyrkou

KIOS Research Center for Intelligent Systems and Networks

University of Cyprus

University of Cyprus

IEEE STUDENT BRANCH UNIVERSITY OF CYPRUS

IEEE Computational Intelligence Society Cyprus Chapter

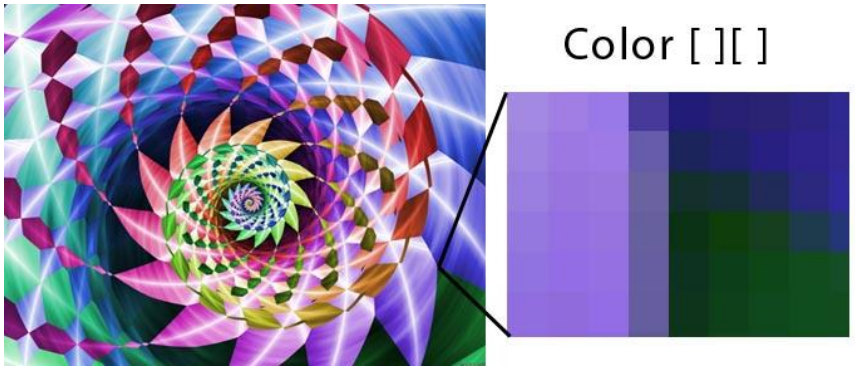
KIOS Center for Intelligent Systems & Networks

Introduction

- How cool it would be to have image recognition on your Android phone?
- A lot of software available in C, C++, and Java for Android
- Neuroph provides a simple API to work with neural networks
- Since version 2.6, Neuroph provides a compatibility layer to run image recognition on Android
  - Libraries loaded as .jar files
  - Code in JAVA
- Note that this is a very basic example, a kind of 'Hello World' for image recognition
  - But if you get deeper into this you might be able to get more advanced stuff.

Image Processing Fundamentals

- An image is a 2D array of colors
- Each array position is a tuple of three basic colors: red, green and blue (RGB).
- Each color can be represented as a combination of these three colors.
- When all colors have the same value the result is a grayscale value
- By thresholsing the grayscale value we get a black and white image



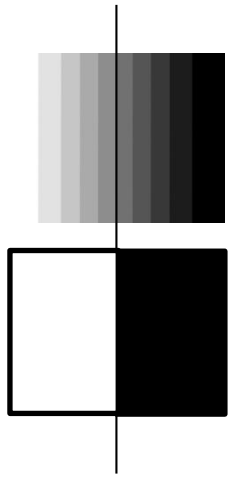
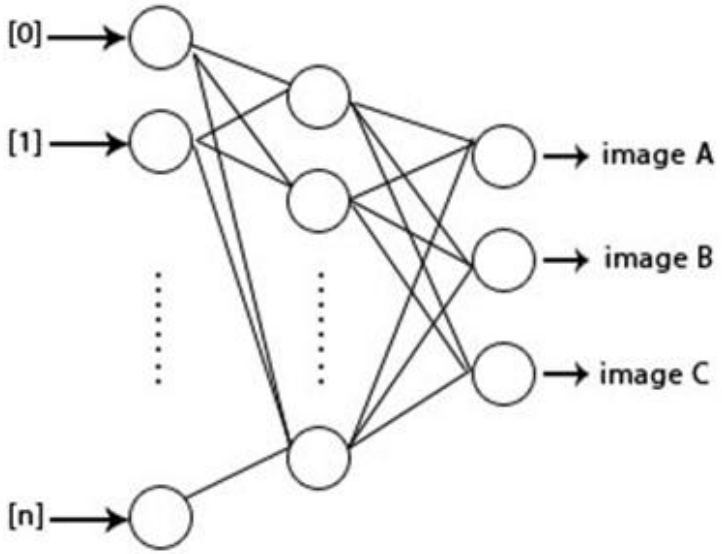


Image Recognition with Neural Networks

- Create a single one-dimensional array so it contains all red values, then all green and at the end all blue values.
- Feeding the image vector to a multi layer perceptron.
- Each input neuron corresponds to a vector element.
- Each output neuron corresponds to one image or image class. So if network output is [1, 0, 0] that means that input is recognized as 'image A'.



Challenges of Recognition Using Video

- Variations in object size, shape, orientation
- Changes in Brightness
- Shadows
- Noise
- Object location within the image
- Many More…

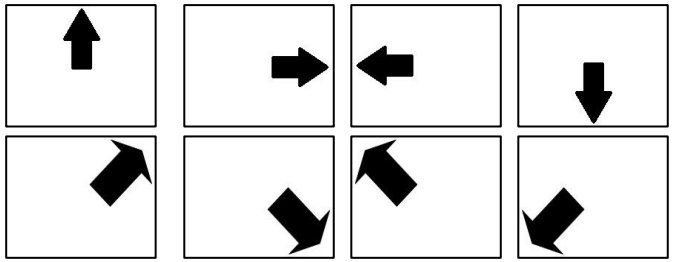
- Complex Feature Extraction
- Image Preprocessing

Neural Network for Image Recognition

- Neuroph Studio
  - Create Neuroph project
  - Create image recognition neural network
  - Train network
  - Test network
  - Save & deploy network for Arrow Recognition

- Numbers:

- Arrows:



The Android Application

- Initial Step:
  - Download and run a simple camera application
- Step 1:
  - Add some simple image processing
  - Run the app again with the new output
- Step 2:
  - Integrate machine learning to Android application
  - Do recognition using device camera

Application Interface

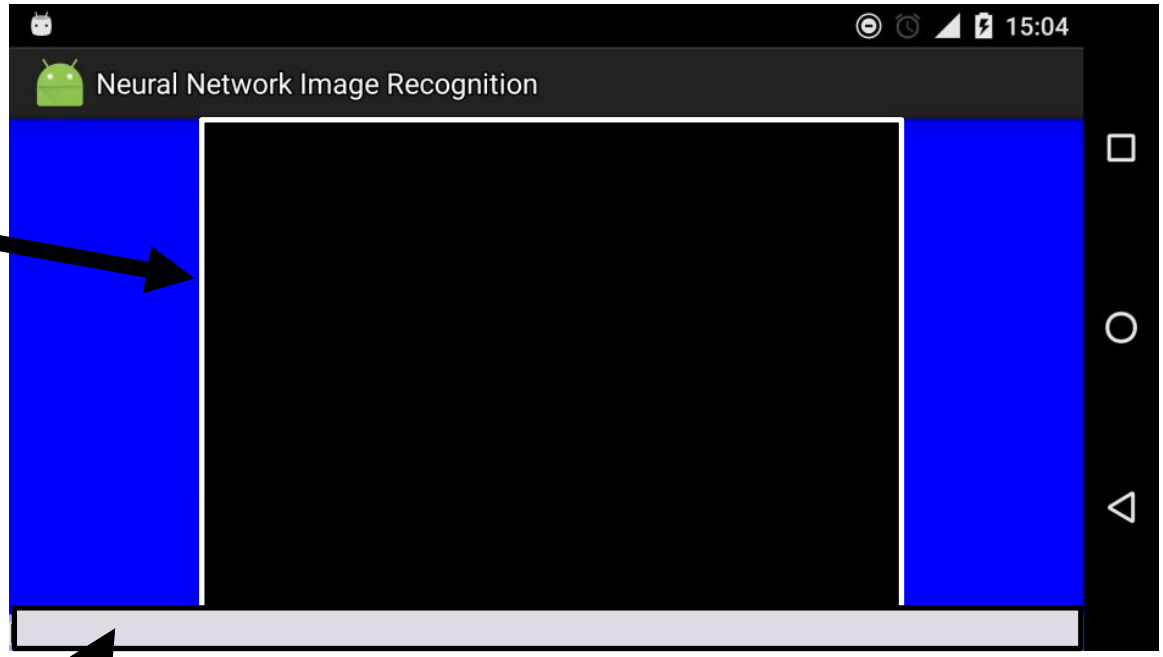
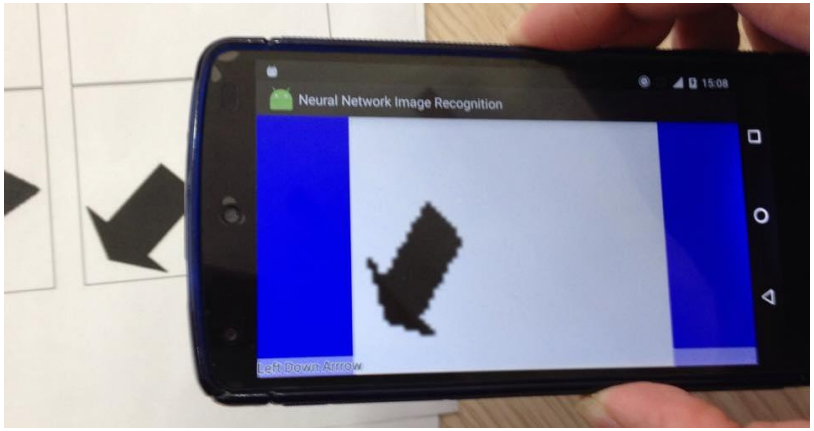


Image View

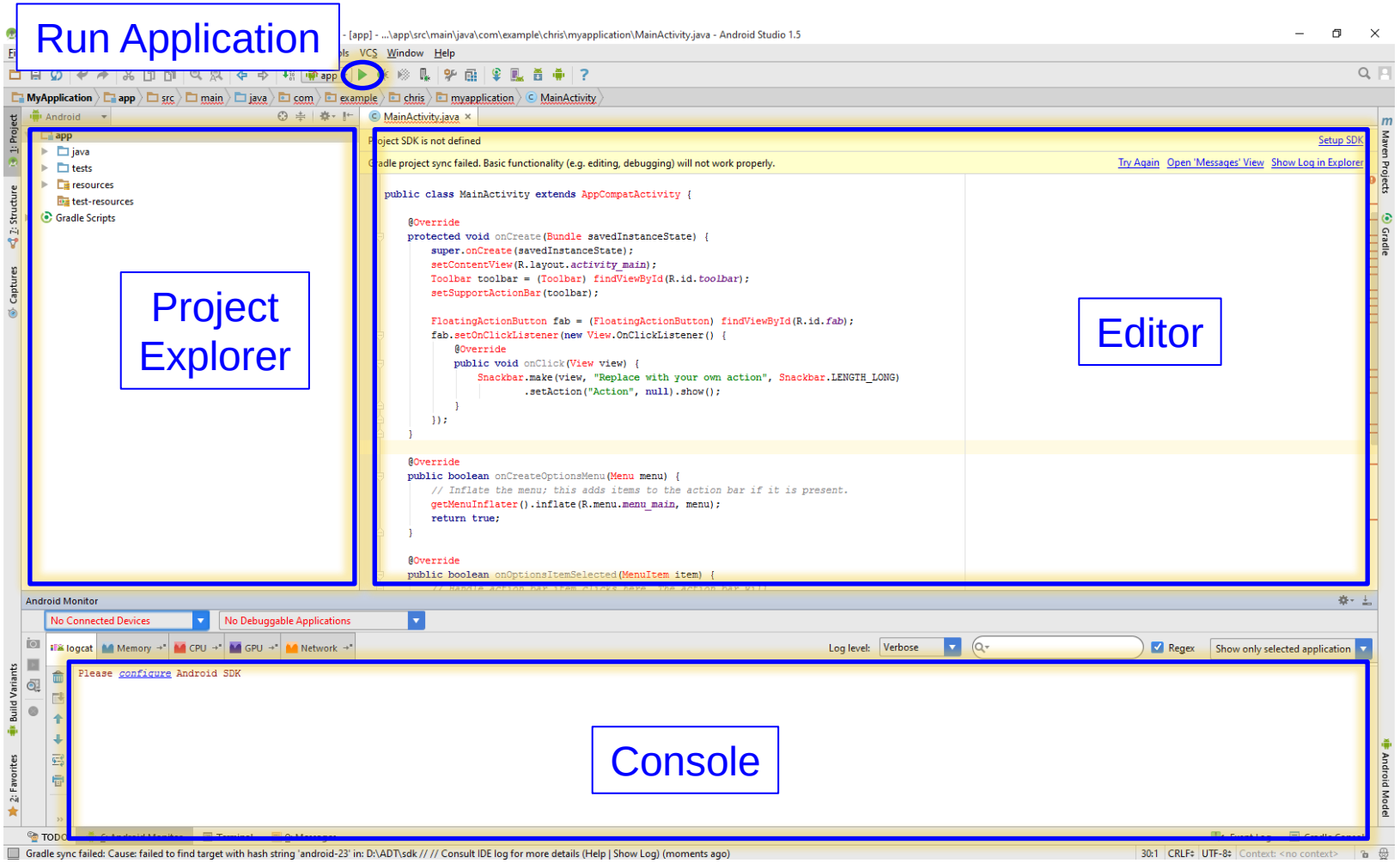
Displays the captured image

Text View

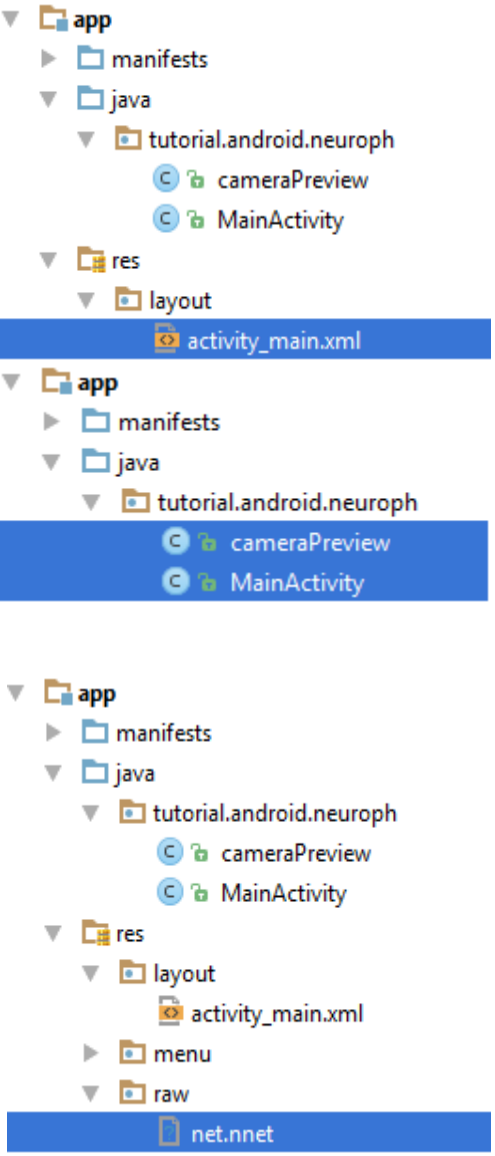
Displays the network output



Android Studio Overview



Components of The Android Application



- <Project Path>/app/manifests
  - Mandatory file for all android applications
  - Declares the activities to run
  - Supported Android Versions
  - Permissions (Camera, I/O)
- <Project Path>/app/res/layout/activity_main.xml
  - A layout defines the visual structure for a user interface, such as the UI
- <Project Path>/app/src/main/java/MainActivity.java
  - Contains the starting activity
  - Get control of UI objects
  - Start other activities
- <Project Path>/app/src/main/java/cameraPreview.java
  - Handles connectivity with camera and the processing necessary for the image processing and machine learning
- <Project Path>/app/res/raw/<network_name.nnet>
  - The folder that contains the neural network model