Deep Learning for Advanced Robot Perception

RBE 595

Fall 2016

Autonomous Navigation of Drones in an Indoor Environment

Shanmuga perumal

Problem Statement

To develop a deep learning system that can navigate a GPS deprived, indoor environment without running into obstacles autonomously

Applications

- Warehousing
- Security
- Entertainment
- Search and rescue (corridors)

Hardware Used and Software Used

DJI Matrice 100 with Intel Realsense

DJI SDK ROS package and Realsense ROS package

- DJI's package to control drone
- Used SDK to write scripts that map keyboard commands to drone motion
- Realsense package has nodes that publish the required information on different topics

Parrot ARdrone

Parrot ARdone

- Front facing camera
- Downward facing camera
- Downward facing Ultrasonic sensor
- Has a ros package
- Cheap and small

However cannot carry additional payload,short battery life

Keras

- High level neural network library in python
- Supports Theano and Tensor flow
- Easy to understand and use
- Fast development

ROS

- Framework for writing robot software
- Modular

ROS Bag

- Gives us the means to record certain or all topics in a file called a ros bag,which can be played back later
- Image_view package to convert ros images to jpgs

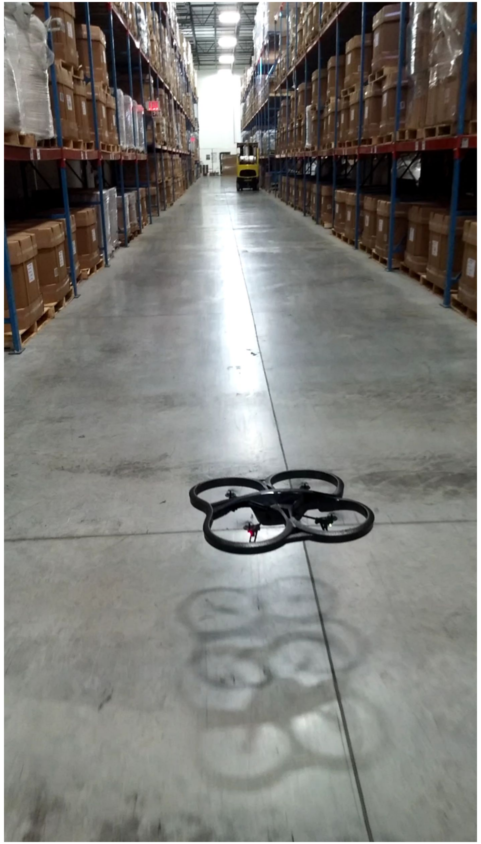
Data set

Available datasets

- Technical University of Munich has a RGB-D data set that also provides accelerometer data from the depth sensor
- Cornell has a similar data set that uses just RGB images

Data Collection

- Collect IMU data and images
- IMU data noisy
- Use command velocities sent to the drone from a human controller and use linear activation functions or scale sigmoid activation functions
- Discretize the motion into primitives and formulate as a classification problem



New Problem Definition

Design a deep learning classifier that when given an input will predict the probability of a control action to be taken for a set of five possible actions.

- **The five possible actions are:**
    - **Go straight**
    - **Turn left**
    - **Turn right**
    - **Slide left**
    - **Slide right**

Data Collection

- Recorded data for each class separately by hand
- Include all ranges of motion that the drone might encounter
- 4115 images in total

Difficult to get access to a warehouse

Network

VGG19

- Very deep convolutional networks
- 3x3 filters with a stride of 1
- 19 weight layers
- Modified the last layer
- Loaded pretrained weights for vgg19 as initialization
- Trained all the fully connected layers on the data set collected
- 98.17% validation accuracy

Test

- Turn_right:0.768

Test

- Go straight:0.999

- Slide_left:0.421
- Go_staright:0.387

Test

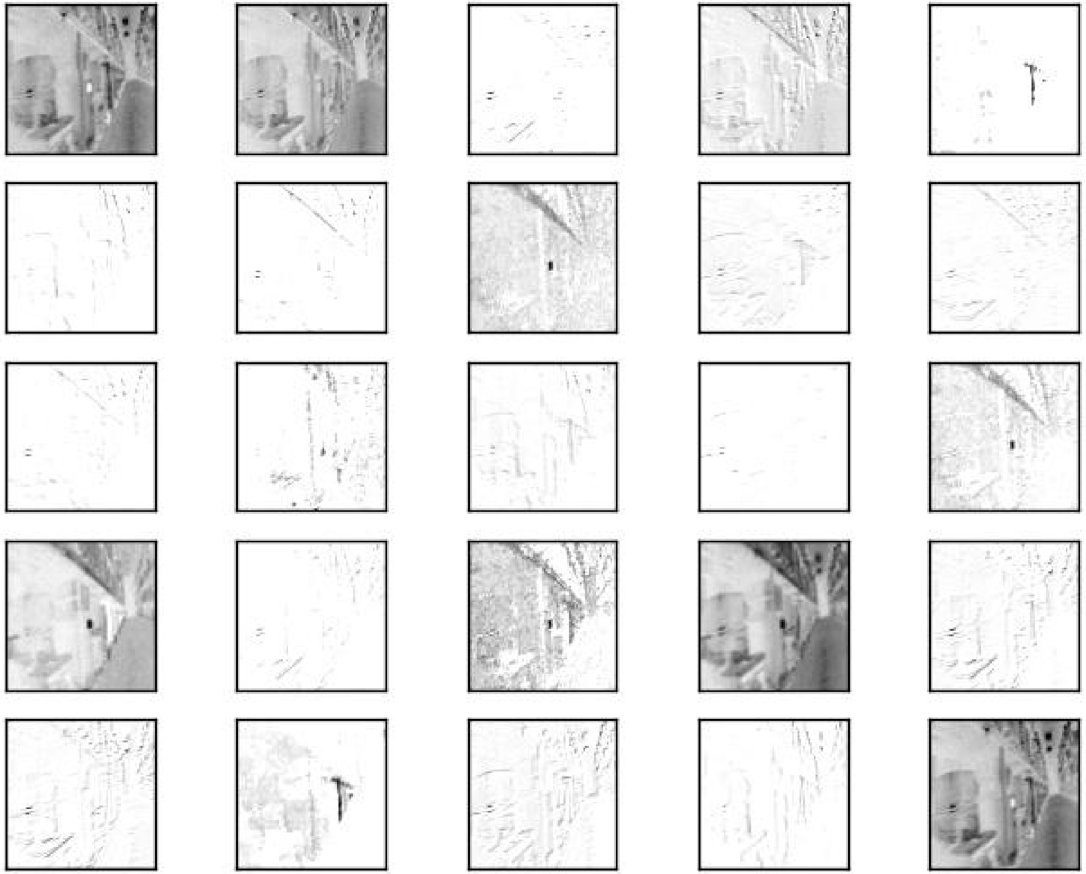
- Slide_left:0.86

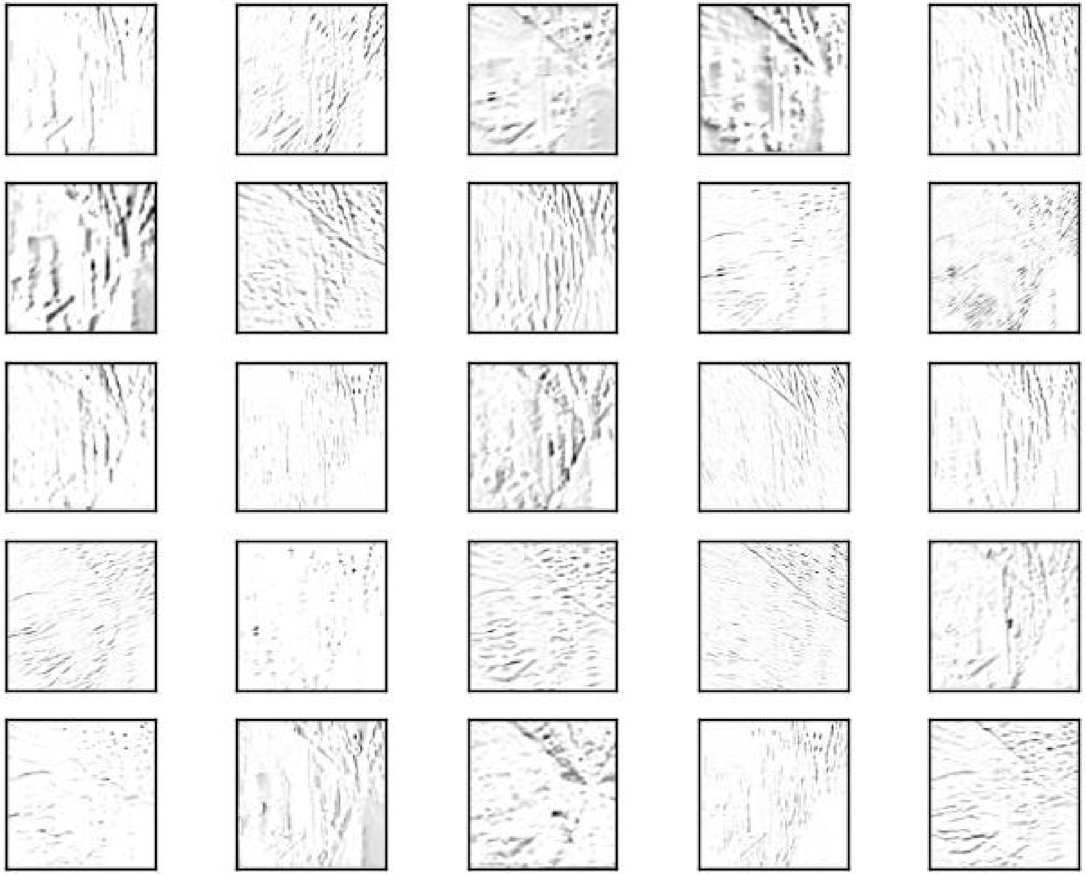
Test

- Slide_right:0.86

Analysis

Image 1





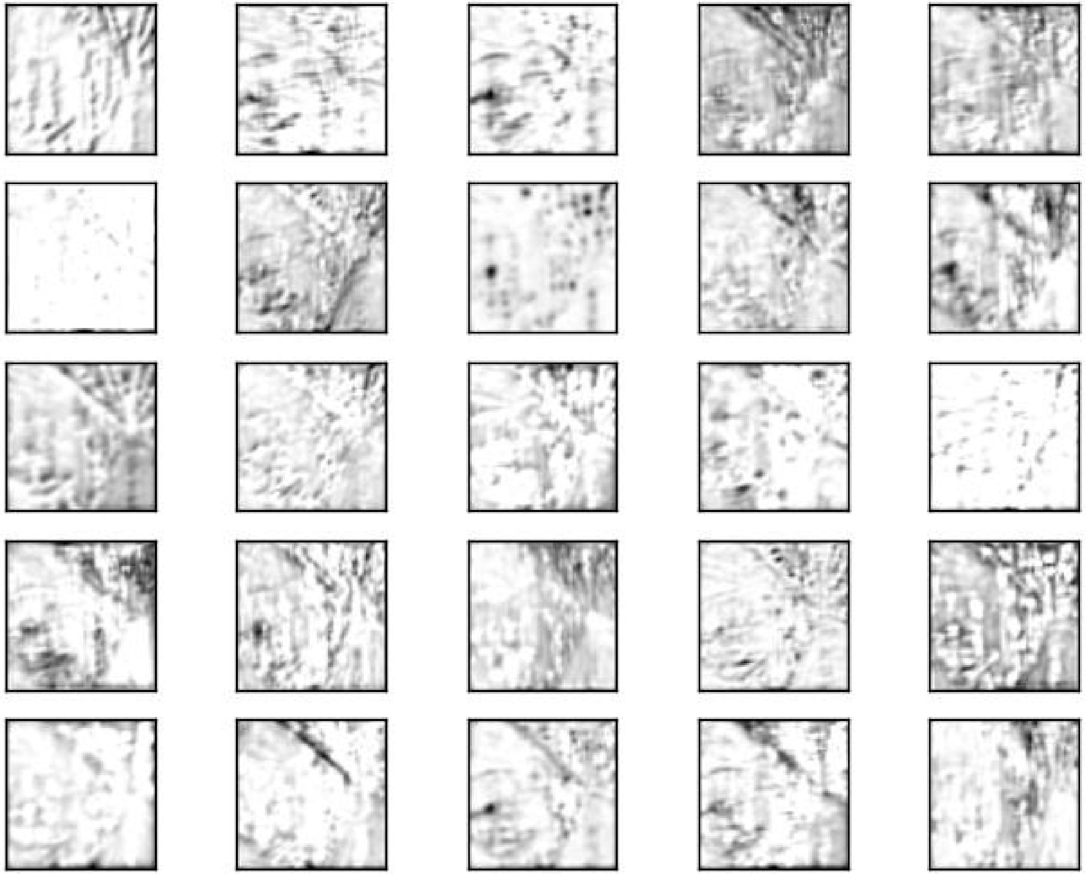
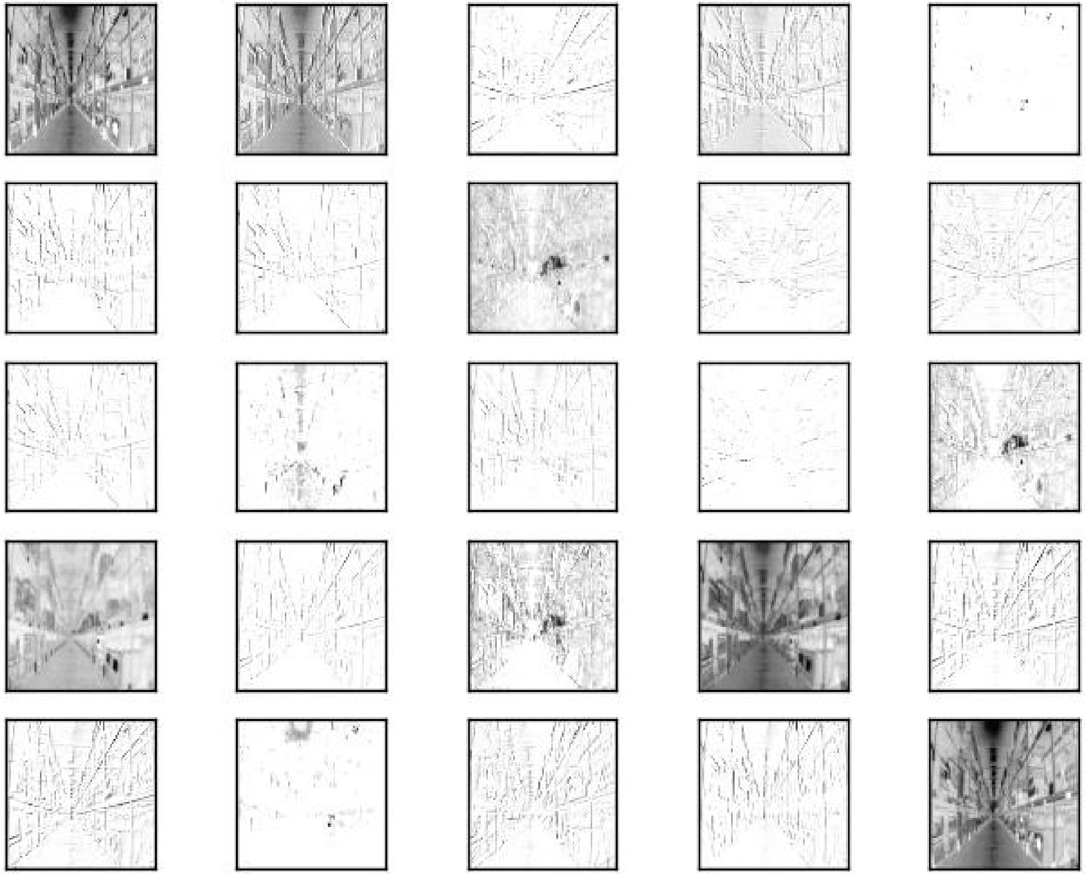
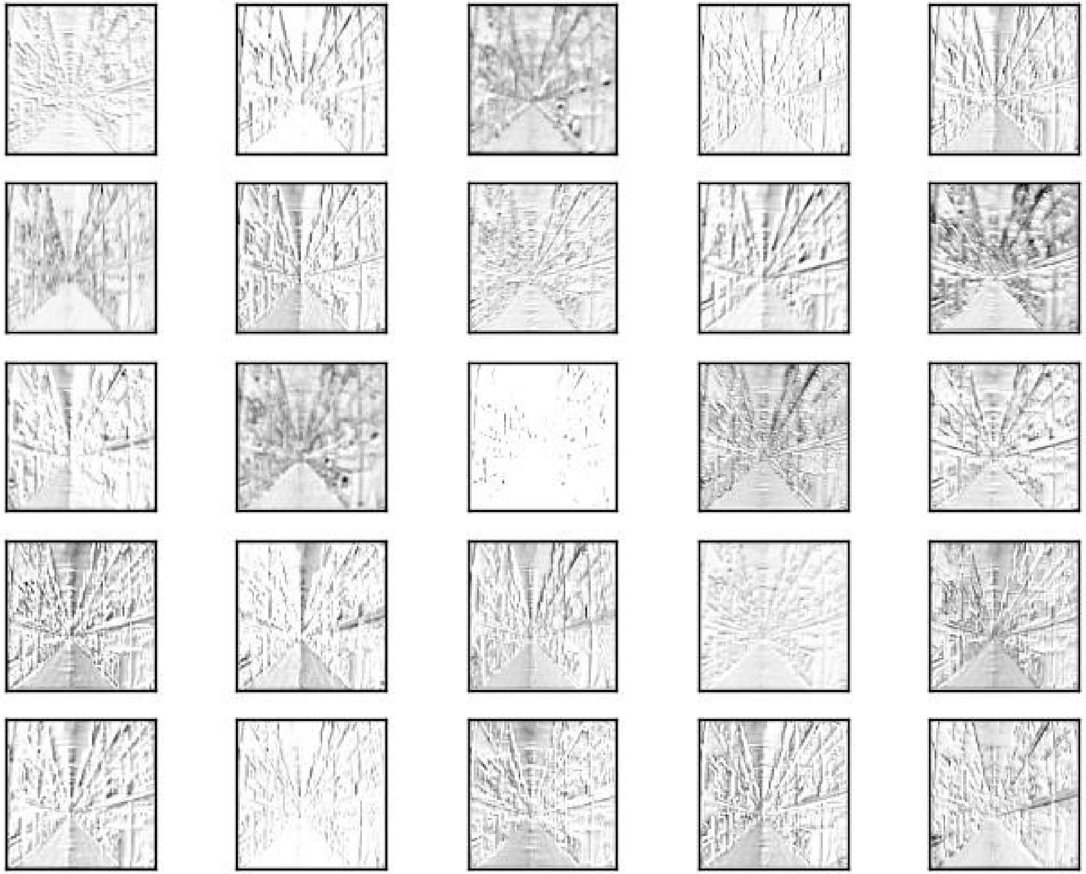
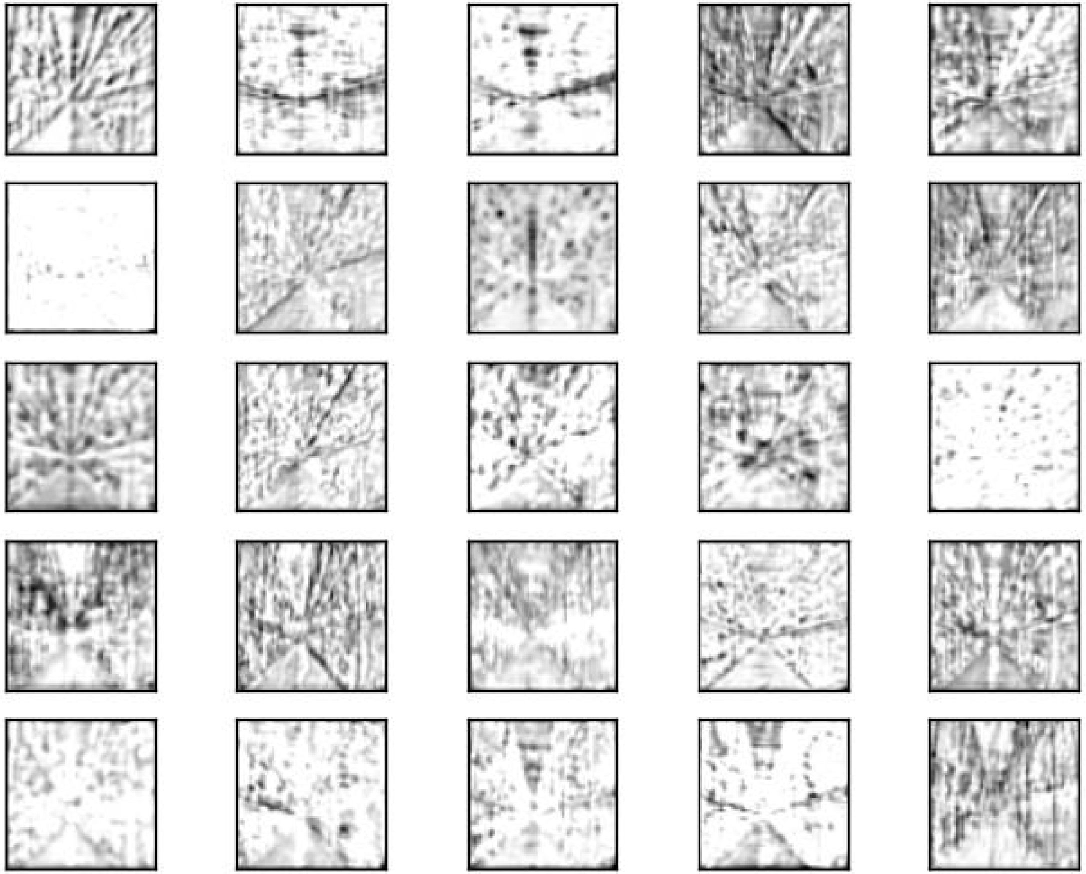
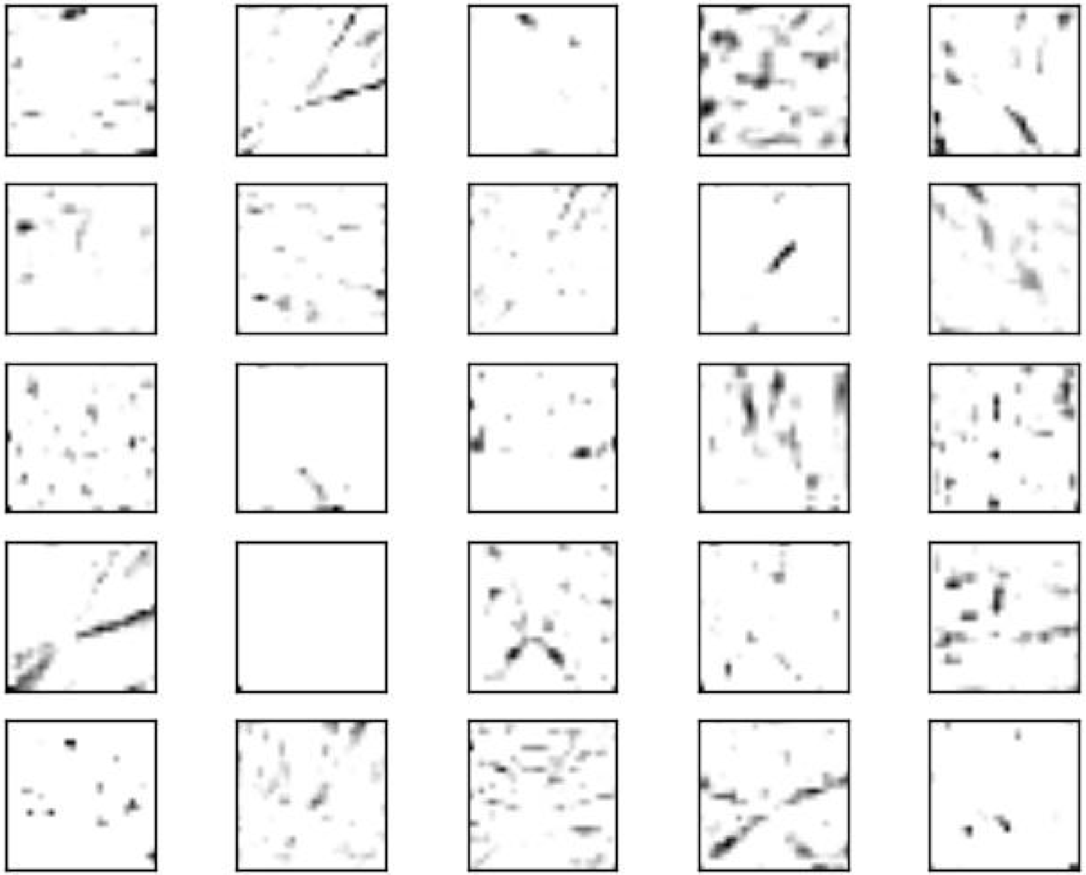


Image 2









Sending commands from the predictions

- Can send the classes as discrete commands with constant velocities .for eg,slide left with 0.5m/s
- Or the the difference between the confidence of slide_left and slide_right is used as a scale factor to multiply with sidewise velocity
- Similarly difference between turn left and turn right can be used for yaw
- Go straight can be used as forward velocity scale factor

Summary of Tasks completed

- Collected data
- Researched and decided on a network
- Trained the network
- Tested the trained network
- Analyse the network
- Code to convert the output of the network to input for the drone

Thank you!