UAV Pose Estimation in Indoor Corridor Using Monocular Vision and Deep Learning

Sachin Verma (216CS1147)
Under the Supervision Of
Dr. Pankaj K. Sa

Department of Computer Science and Engineering
National Institute of Technology, Rourkela

May 26, 2018

Autonomous Navigation

- Devices are capable of riding itself by responding to the "Navigating Environment".
- Fully Programmed
- Generate Flight Command on their own

Autonomous Drone Navigation Scenario

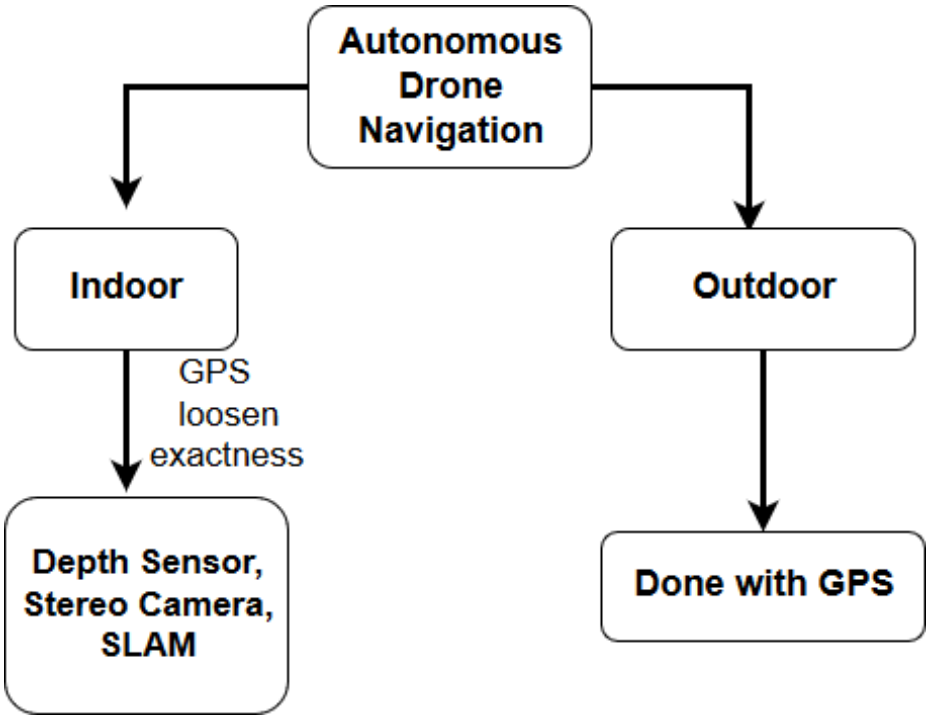


Figure 1: Autonomous Drone Navigation Scenario

Problem Domain

Autonomous Drone Navigation Scenario

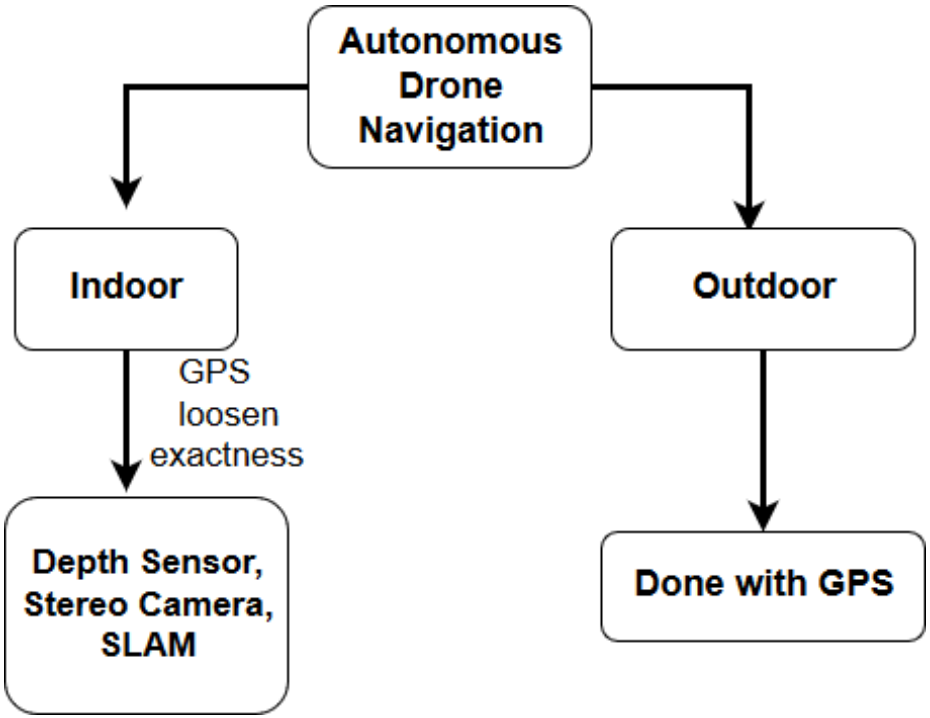


Figure 1: Autonomous Drone Navigation Scenario

Our Working Scenario

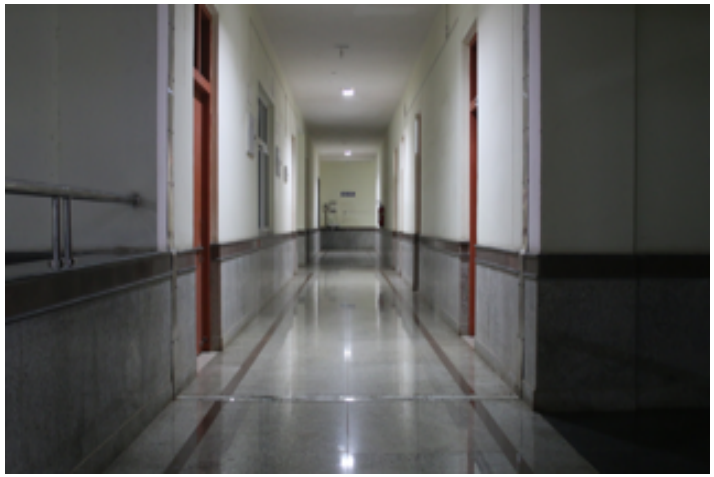


Figure 2: A Corridor in TIIR

Navigation is Modelled as " Regression Problem "

Problem Origin

Problem Origin

Problem Statement

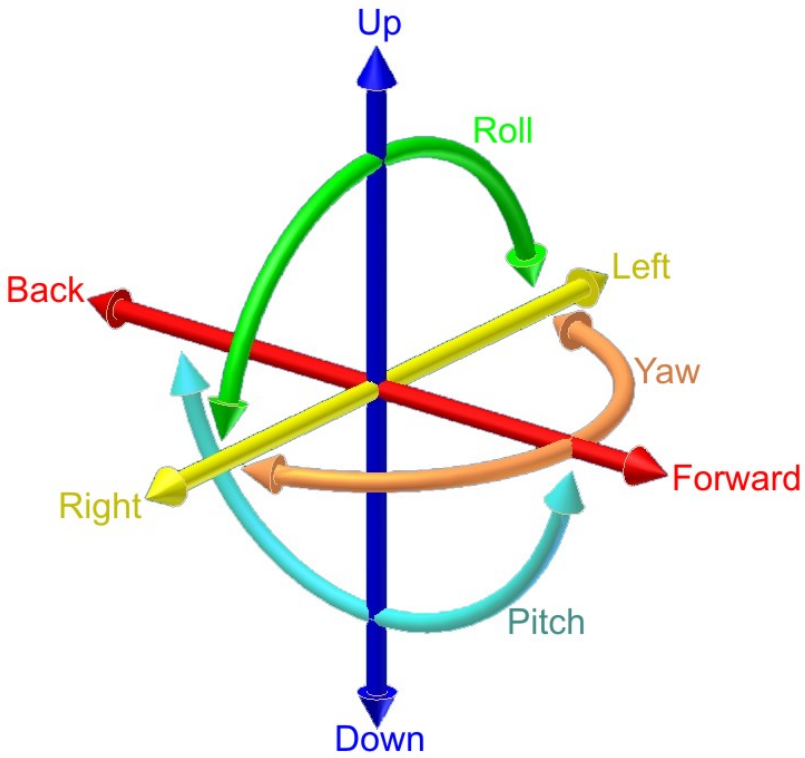


Figure 3: degrees of freedom

Problem Statement

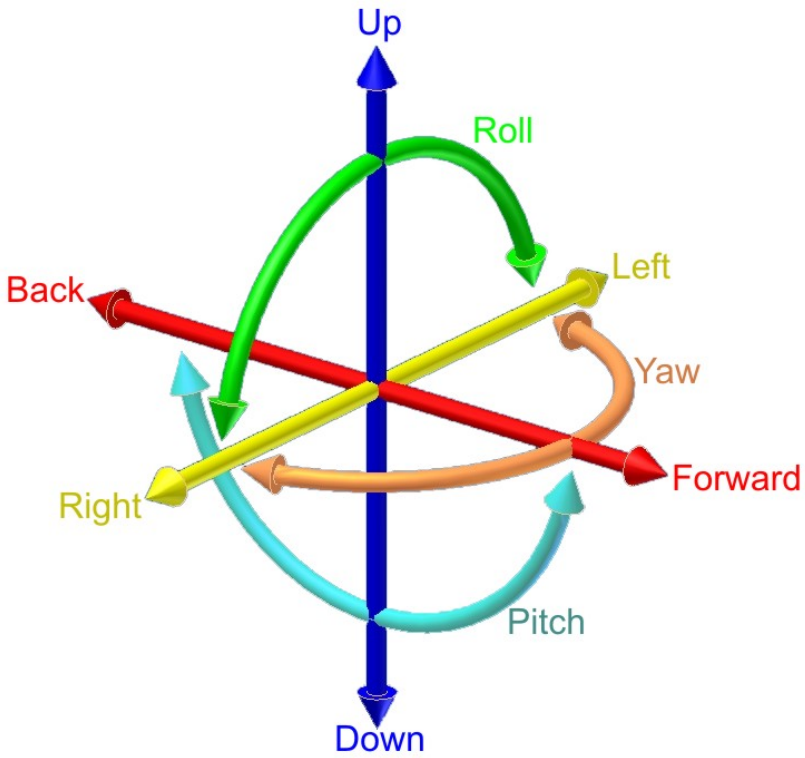
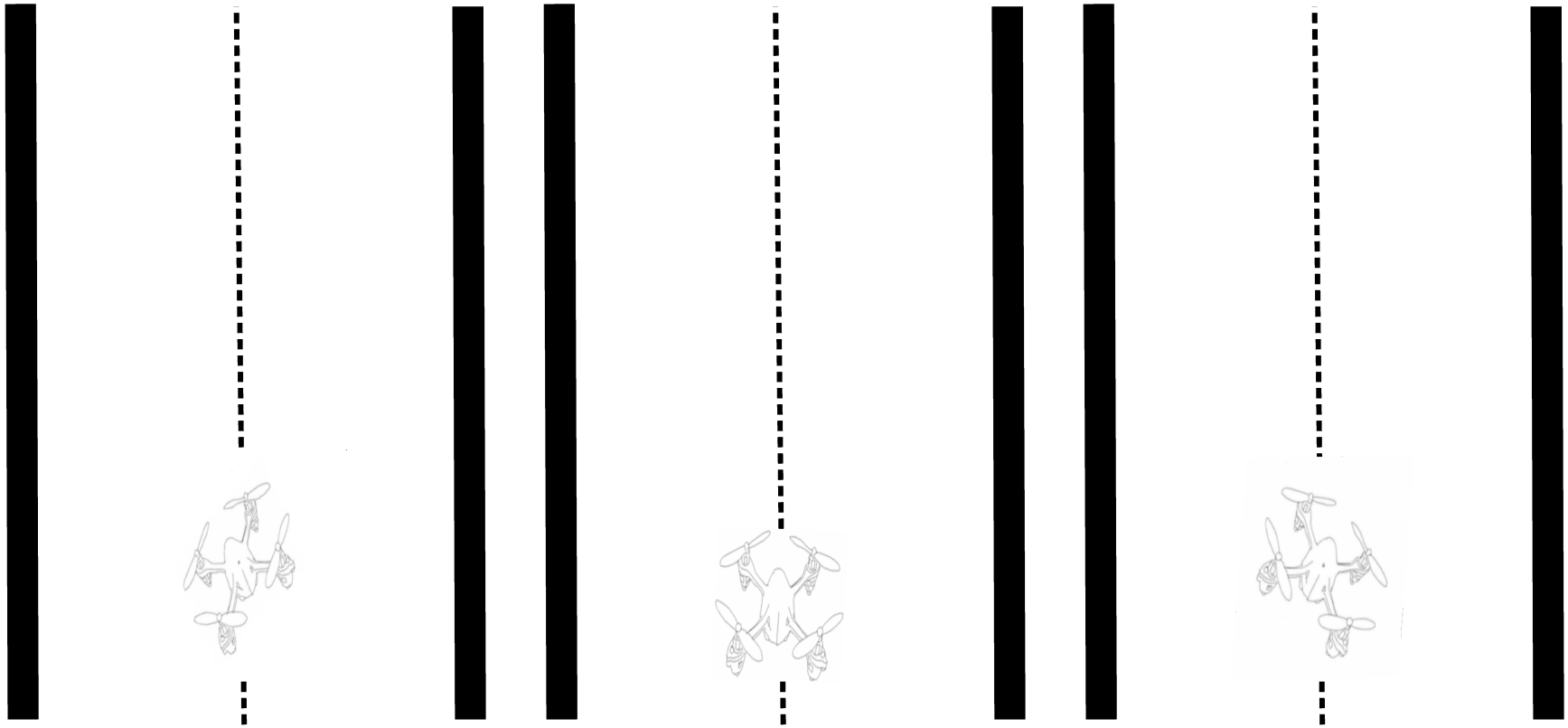


Figure 3: degrees of freedom

Drone pose correction in the indoor corridor environment such that device must always be facing straight when it is at the center of the transit environment using

"YAW FLIGHT COMMAND"

Problem Scenario



Challenges to Drone Navigation/Localization

- Payload
- Power
- Degrees of Freedom
- Stability

Literature Survey

Our Related Study Falls into any of the following

1. **Range Sensor based Navigation** $\rightarrow$ Depth Information

Literature Survey

Our Related Study Falls into any of the following

1. **Range Sensor based Navigation** $\rightarrow$ Depth Information
2. **SLAM based Navigation** $\rightarrow$ 3D Map Generation and Simultaneous Mapping

Literature Survey

Our Related Study Falls into any of the following

1. **Range Sensor based Navigation** $\rightarrow$ Depth Information
2. **SLAM based Navigation** $\rightarrow$ 3D Map Generation and Simultaneous Mapping
3. **Stereo Vision based Navigation** $\rightarrow$ Two Lens

Literature Survey

Our Related Study Falls into any of the following

1. **Range Sensor based Navigation** → Depth Information
2. **SLAM based Navigation** → 3D Map Generation and Simultaneous Mapping
3. **Stereo Vision based Navigation** → Two Lens
4. **Navigation With Learning based Algorithm** → Classification of Follow Command

Literature Survey

Our Related Study Falls into any of the following

1. **Range Sensor based Navigation** → Depth Information
2. **SLAM based Navigation** → 3D Map Generation and Simultaneous Mapping
3. **Stereo Vision based Navigation** → Two Lens
4. **Navigation With Learning based Algorithm** → Classification of Follow Command
5. **Deep Learning**

Literature Survey Cont.

- **ALVINN(NIPS, 1989)**
  - For land vehicle
  - 3-layer neural network
  - 99% accuracy

- **Quadrotor using Minimal Sensing For Autonomous Indoor Flight.(EMAV, 2007)**
  - one ultrasonic sensor and four infra-red sensors.
  - fully autonomous flight
  - most of publicly available quadcopters doesn't have infra red sensor.

Literature Survey Cont.

- **AlexNET(NIPS, 2012)**
  - Reduced Top-5 Error from 26.2% to 15.4%
  - 5 Convolutional Layer and 3 Fully Connected layers
  - Extensive Data Augmentation

- **Monocular Vision Slam for Indoor Aerial Vehicles(JECE, 2013)**
  - Used Monocular Camera
  - computationally expensive due to the 3-D reconstruction

Literature Survey Cont.

- **ZFNet(ECCV, 2013)**
  - Fine-Tuning of Alexnet
  - Developed a visualization technique named Deconvolutional Network

- **VGGNet(arXiv, 2014)**
  - Used only 3x3 filters
  - Decrease Spatial dim. but increase depth of the feature.

Literature Survey Cont.

- **InceptionNet(CVPR, 2015)**
  - Parallel convolutions
  - no fully connected layer
  - First model to Concatenate the outputs

- **Deep Neural Network for Real-Time Autonomous Indoor Navigation (arXiv, 2015)**
  - Used classification for flight command
  - No tilted case considered

Literature Survey cont...

- **ResNet(CVPR, 2016)**
  - Came up with the concept of Residual Block
  - Gradient flow is easy

- **DenseNet(CVPR, 2017)**
  - dynamic model which connect every layer with all of its previous layer
  - Fast Gradient Update

Research Motivation

- Weak strength of GPS[1] signal in indoor environment.
- Existing methods require much hardwares and software dependencies.
- Lacking Pilot's View

Objectives

- **Corridor Mid-Point Estimation**
- **Drone Alignment Estimate**

Best Case for Navigation

Proposed Scheme

AT MIDDLE OF THE CORRIDOR

FACING STRAIGHT

Flight Command → MOVE FORWARD

Proposed Scheme Cont.

AT MIDDLE OF THE CORRIDOR

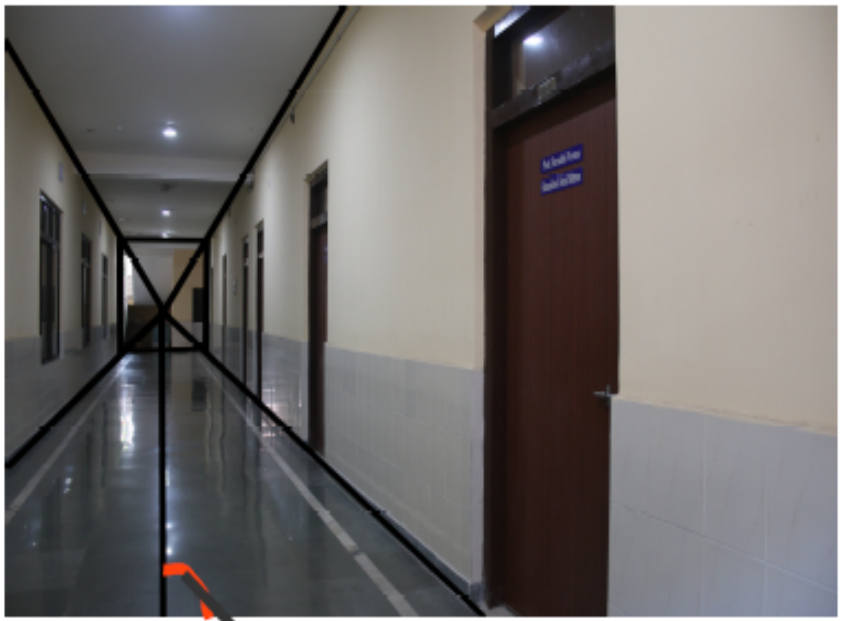
TILTED TOWARD LEFT

Flight Command → YAW RIGHT

Proposed Scheme Cont.

AT MIDDLE OF THE CORRIDOR

TILTED TOWARD RIGHT



Flight Command → YAW LEFT

Localization Algorithm

Algorithm 1: TiltPose →returns pose of the drone

Input: **IMAGE** taken from the onboard camera

Result: flight command to localize st drone always face stright

```
1 d=DistFind(IMAGE)
2 if (d < image.width/2 − 20) then
3 |   return tilted right;
4 else if (d > image.width/2 + 20) then
5 |   return tilted left;
6 else
7 |   return facing straight;
8 |   ;
```

Localization Algorithm

Algorithm 2: DistFind →Distance of point of intersection between the Bisector and Horizontal Axis

Input: **IMAGE** of the the transit environment from Algorithm Tilt-Pose

Result: number of pixels betwwen bottomleft pixel of the image and the intersection point

1 Normalize pixel value of **IMAGE** between 0 to 1;
2 RGB to BGR conversion;
3 Normalize **IMAGE** with mean and standard deviation of ImageNet-10000[2] data set;
4 DistPix = Trained_Model(**IMAGE**);
5 **return** DistPix;

Deep Learning as Solution

- Deep learning class ConvNet[3] enable direct feature extraction over images.
- We learn intrinsic features of the image captured from drone camera to guide pose correction.

Data Set Creation

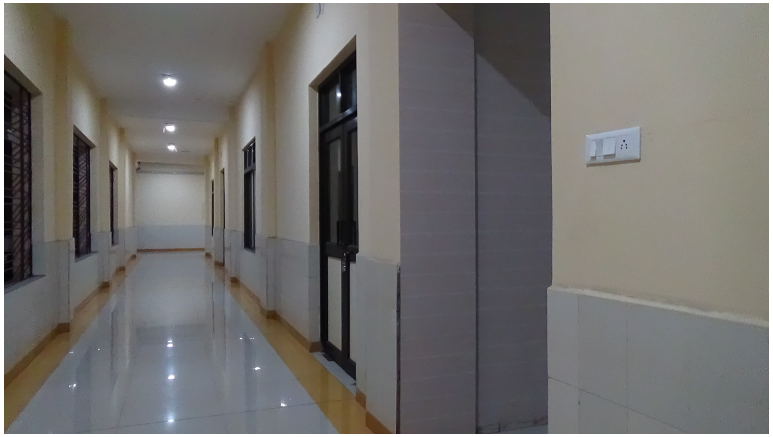
- **107 Corridor** $\rightarrow$ NIT Premises

- **DataSet Size** $\rightarrow$ **65000**
  - 21000 Training Set Image
  - 300 Test Set Image

Data Set Creation (Raw Data)

(a) At Center aligned to Left

(b) At Center aligned to Center



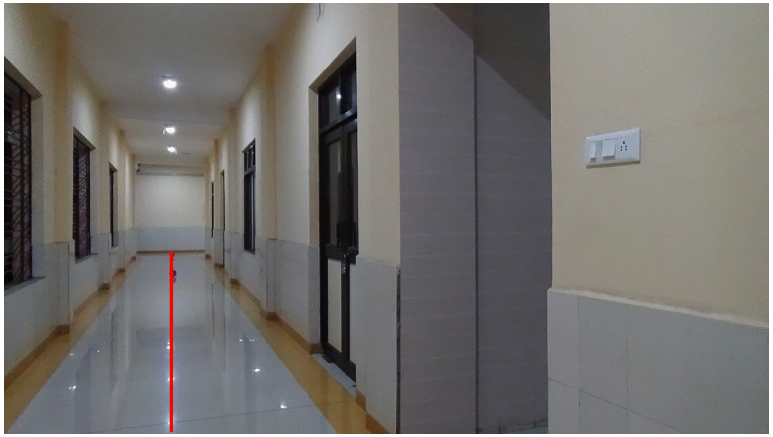
(c) At Center aligned to Right

Figure 4: various instances of indoor corridor captured from onboard

Data Set Creation (Labelled Data) (Cont.)

(a) At Center aligned to Left

(b) At Center aligned to Center



(c) At Center aligned to Right

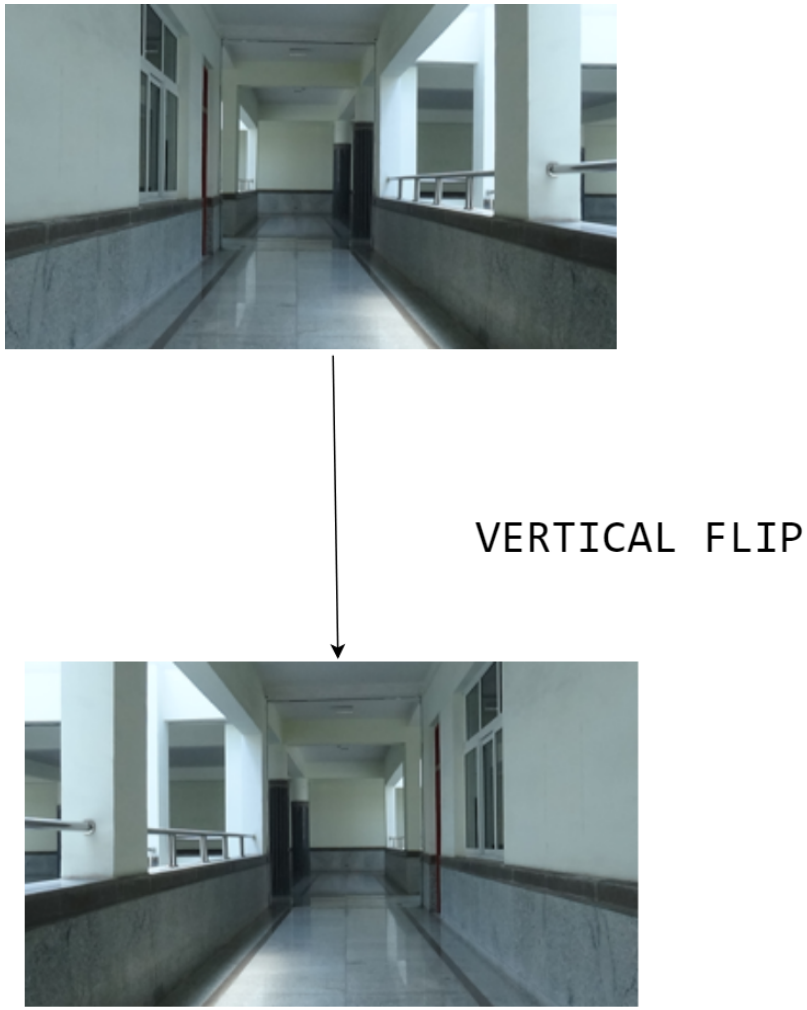
Figure 5: labelled images used to find ground truth

Data Set Samples

Figure 6: various instances of indoor corridors

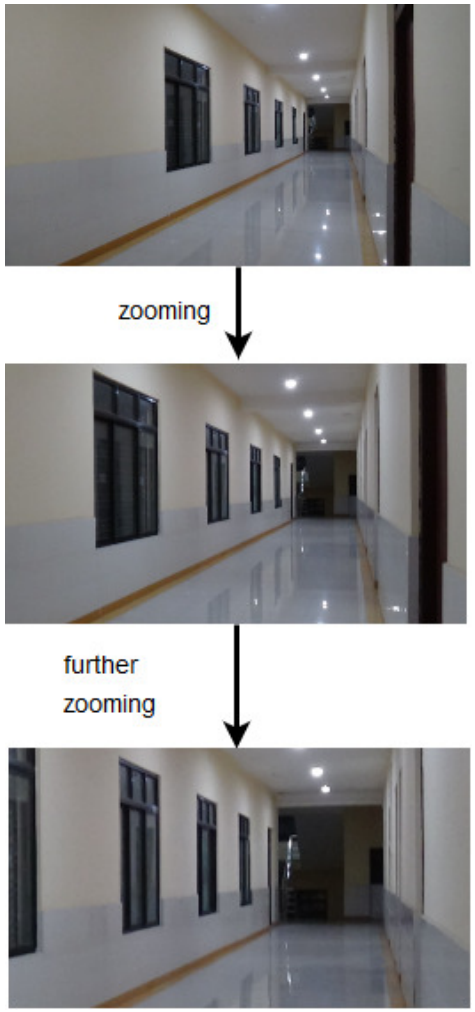
Data Augmentation Technique

- Image Flipping



Data Augmentation Technique (Cont.)

- Image Zooming



CNN Architecture

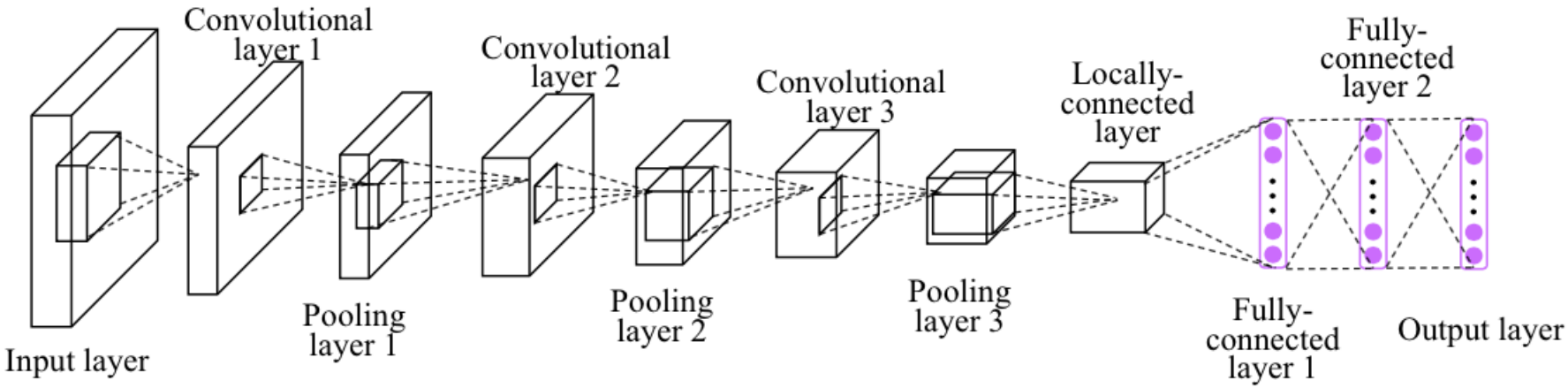


Figure 7: Convolutional Neural Network

CNN Architecture cont..

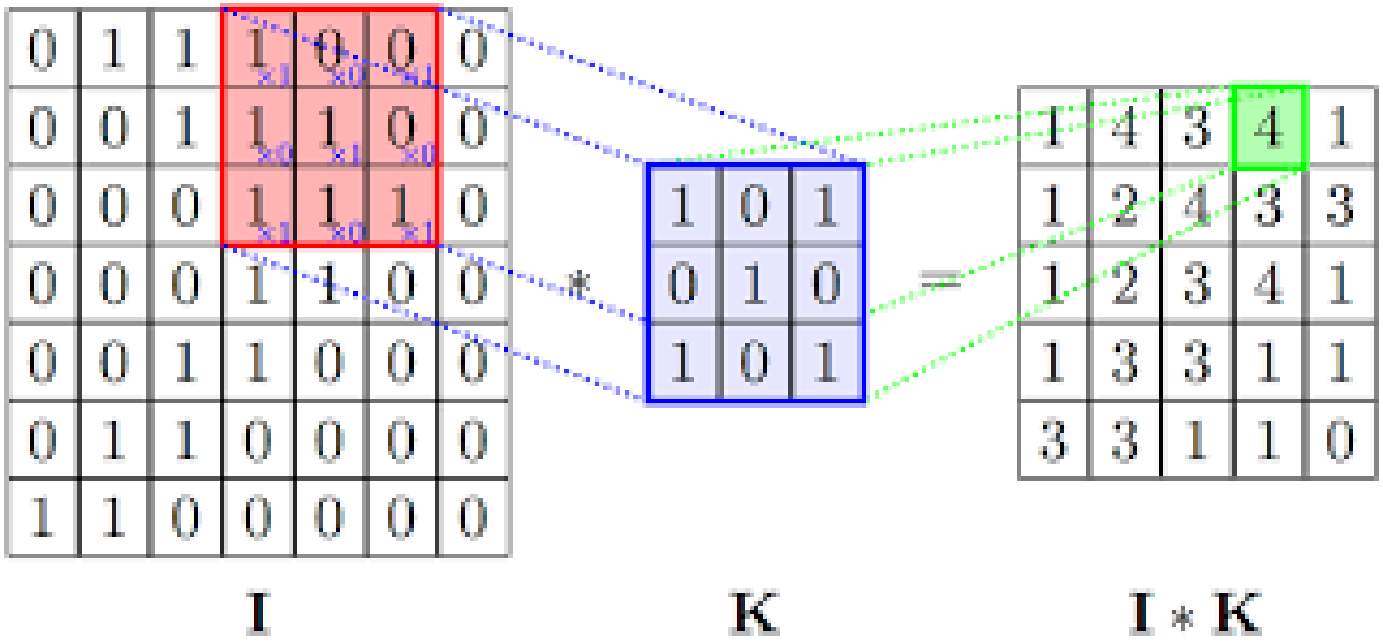
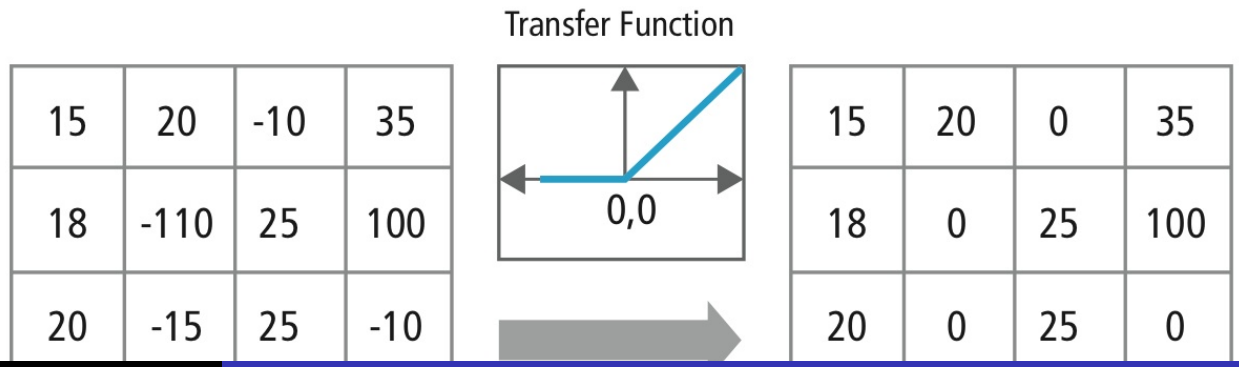


Figure 8: Convolutional Layer



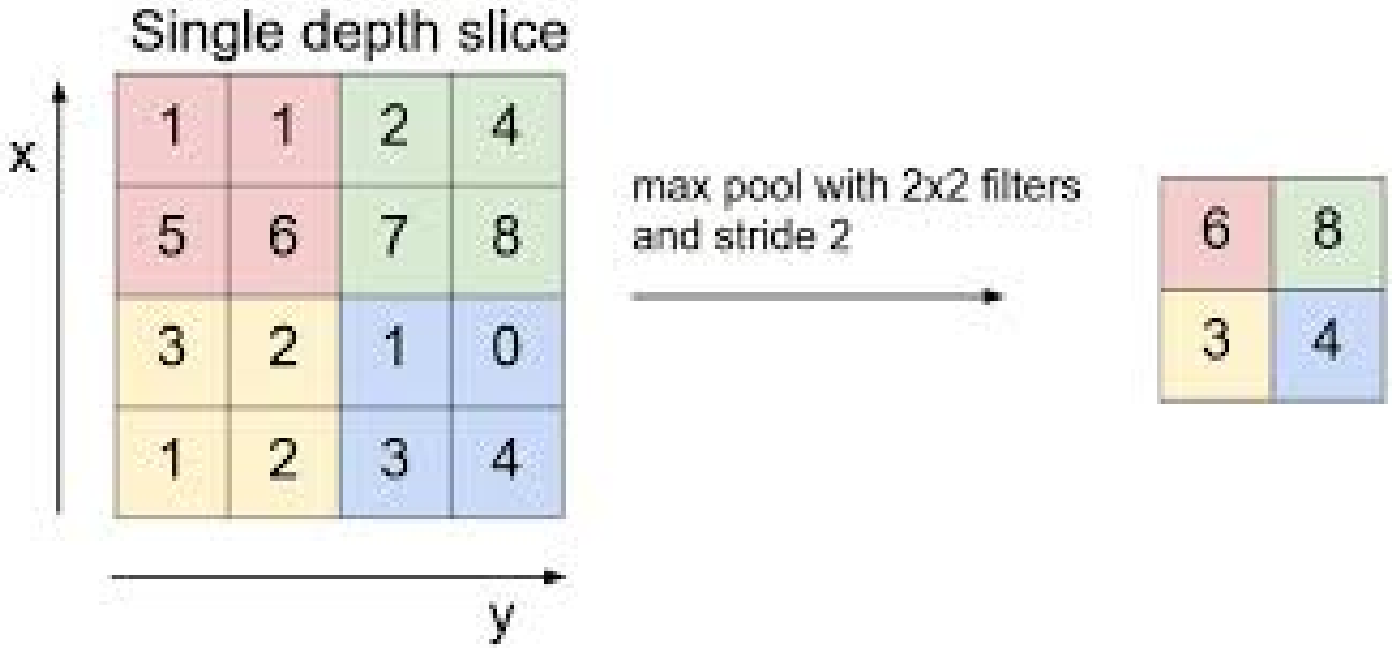


Figure 10: Pooling Layer

Working Model

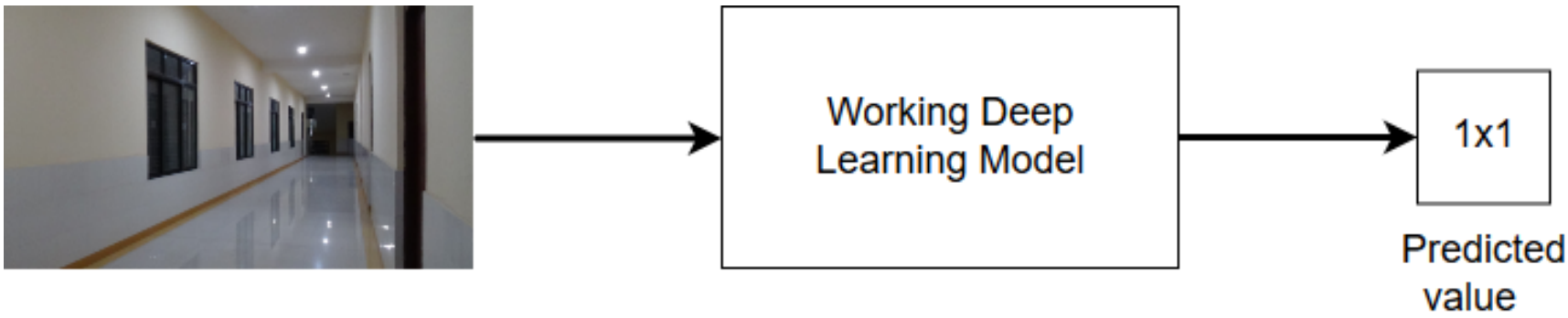


Figure 11: fig explains the working model

Architecture of Working Model

Deep Model Architecture Used			
Pre-Trained Model	**Augmented Layers**	**Output Layer**	**Output**
ALEXNET	-	FC(4096,1)	(1,1)
DenseNet-161	CONV2D(2208,1024,1) CONV2D(1024,128,5) CONV2D(128,16,1)	FC(96,1)	(1,1)
DenseNet-201	CONV2D (1920,1024,1) CONV2D(1024,128,5) CONV2D(128,16,1)	FC (96,1)	(1,1)
INCEPTION-V3	**Main**:CONV2D (1920,1024,1) CONV2D(1024,128,5) CONV2D(128,16,1) **AUX**:CONV2D (768,128,4) CONV2D(128,32,2) CONV2D(128,16,(1,2))	FC (256,1) FC (640,1)	(1,1) (1,1)

Architecture of Working Model Cont...

Pre-Trained Model	**Augmented Layers**	**Output Layer**	**Output**
ResNet-50	CONV2D(2048,1024,1) CONV2D(1024,128,5) CONV2D(128,8,1)	FC(96,1)	(1,1)
ResNet-101	CONV2D(2048,1024,1) CONV2D(1024,128,5) CONV2D(128,8,1)	FC(96,1)	(1,1)
ResNet-152	CONV2D(2048,1024,1) CONV2D(1024,128,5) CONV2D(128,8,1)	FC(96,1)	(1,1)

Training Of Models

- We use Mean Absolute Error function to learn model listed above.

$$Mean\ Absolute\ Error(\hat{y}, y) = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_t - y_t|$$

Here,
$\hat{y}$ is the prediction made on all the training examples
y is ground truth value of all the examples of training set.
$\hat{y}_t$ is the prediction made on t^{th} training example
y_t is ground truth value of t^{th} training set.
n is the batch size.

Convergence Graph for Alexnet

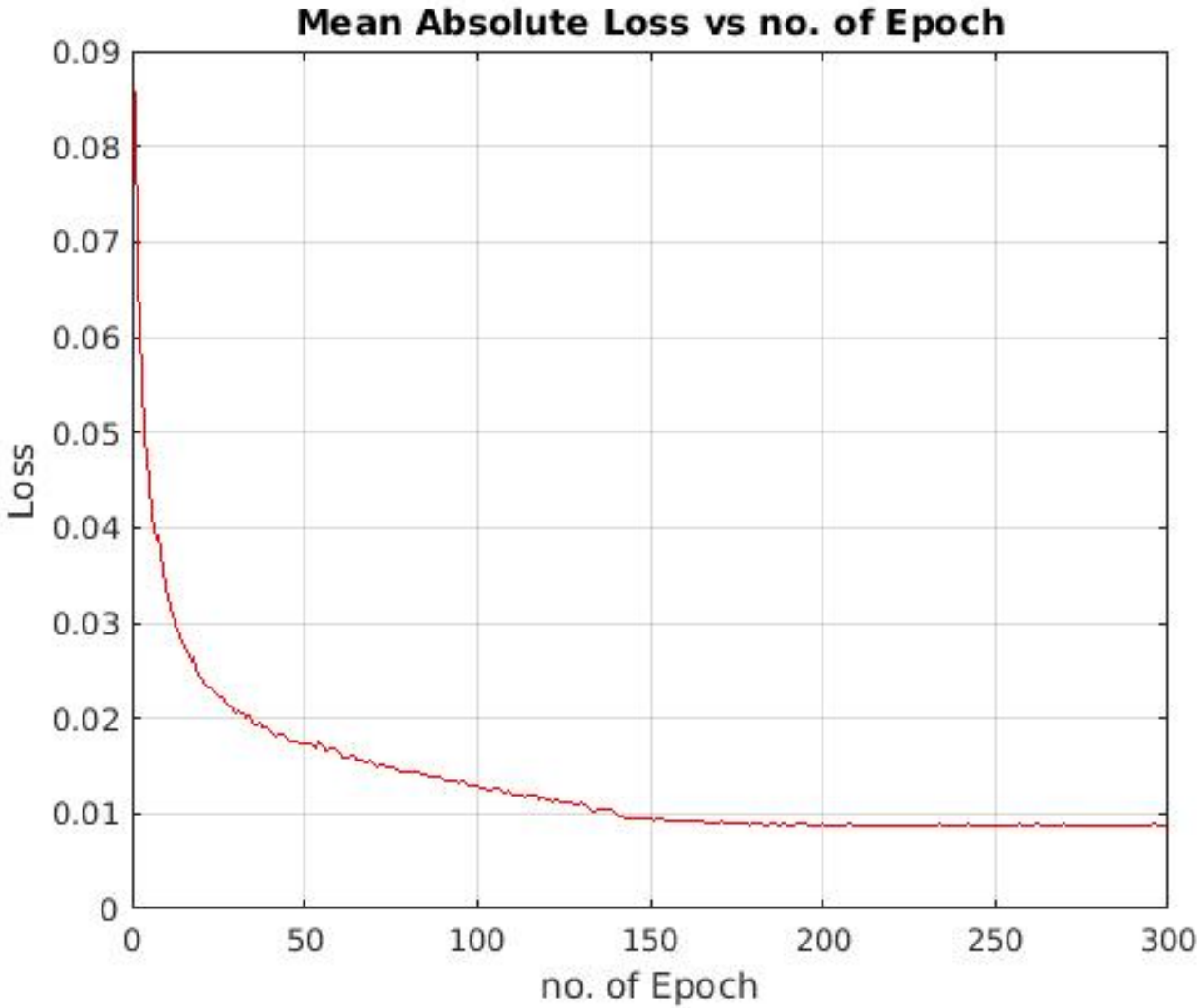


Figure 12: Trianing Loss vs No. of Epoch for MAE for Alexnet

Convergence Graph for DensNet-161

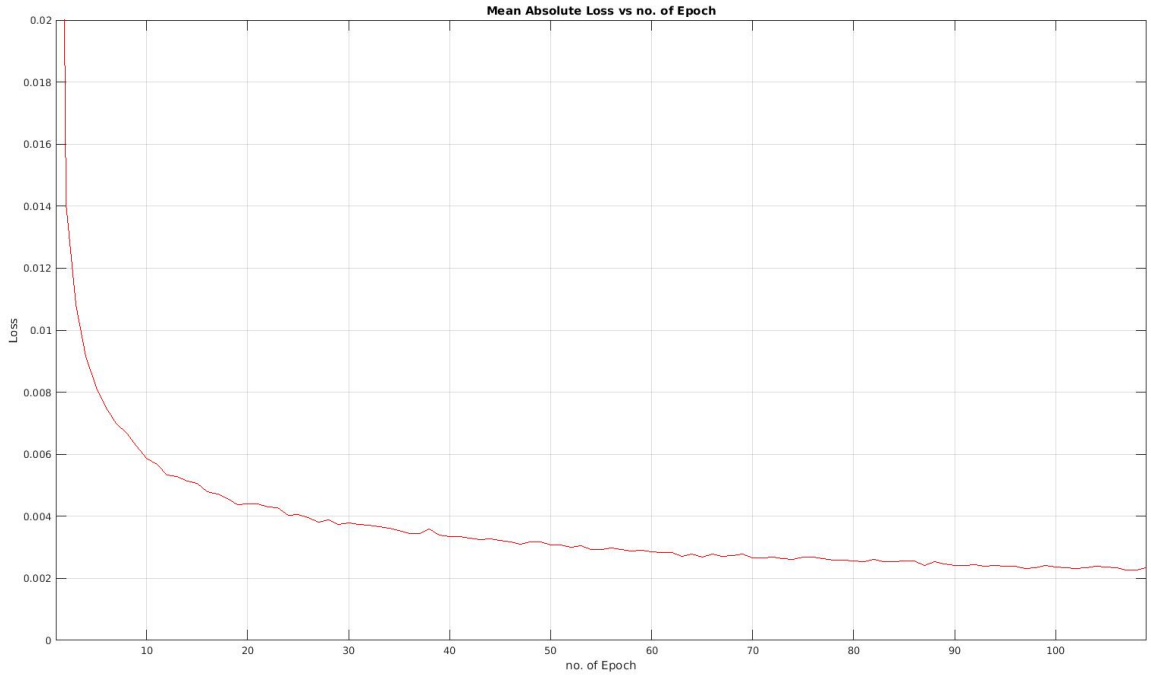


Figure 13: Trianing Loss vs No. of Epoch for MAE for DensNet-161

Convergence Graph for DensNet-201

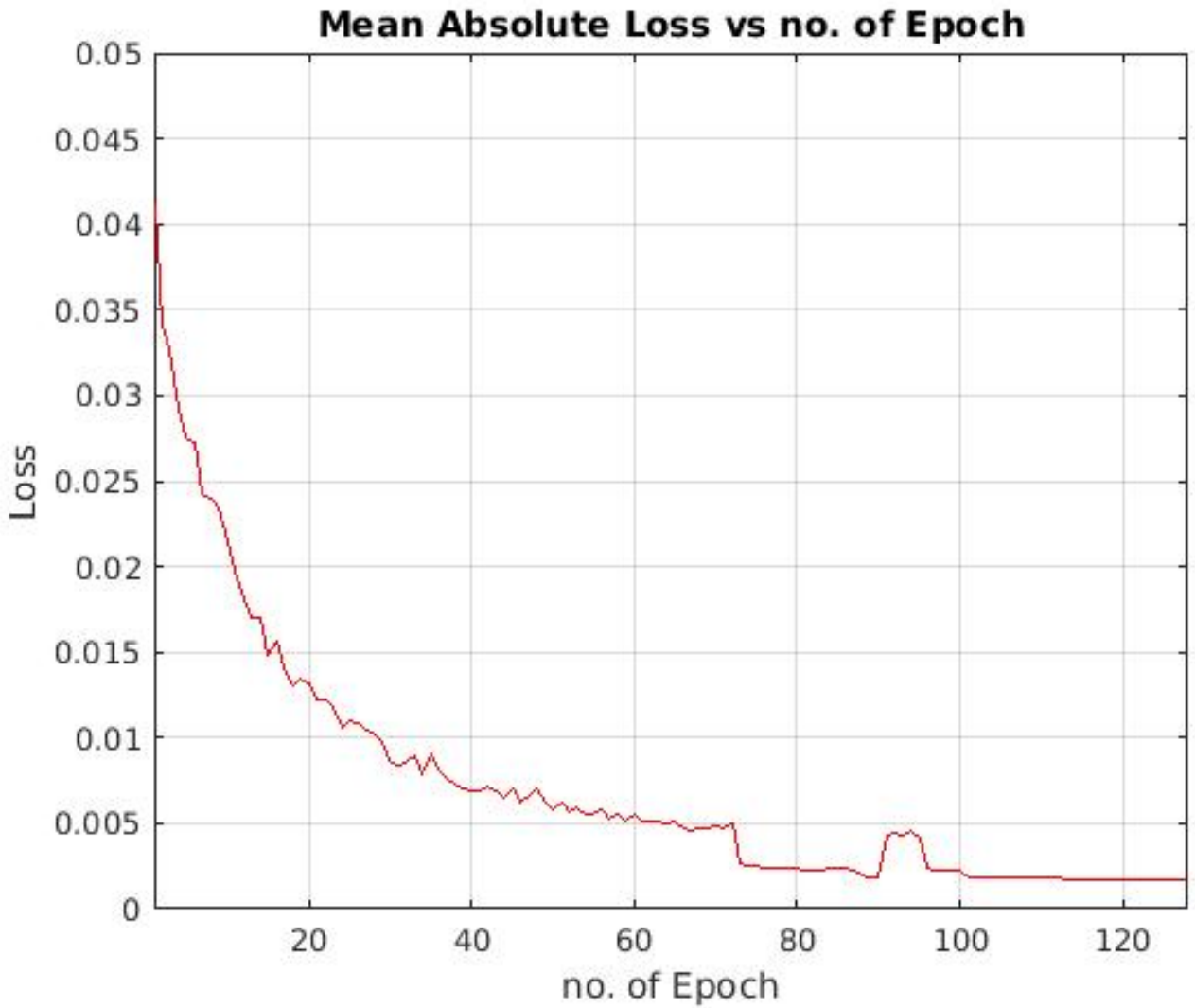


Figure 14: Trianing Loss vs No. of Epoch for MAE for DensNet-201

Convergence Graph for Inception-V3

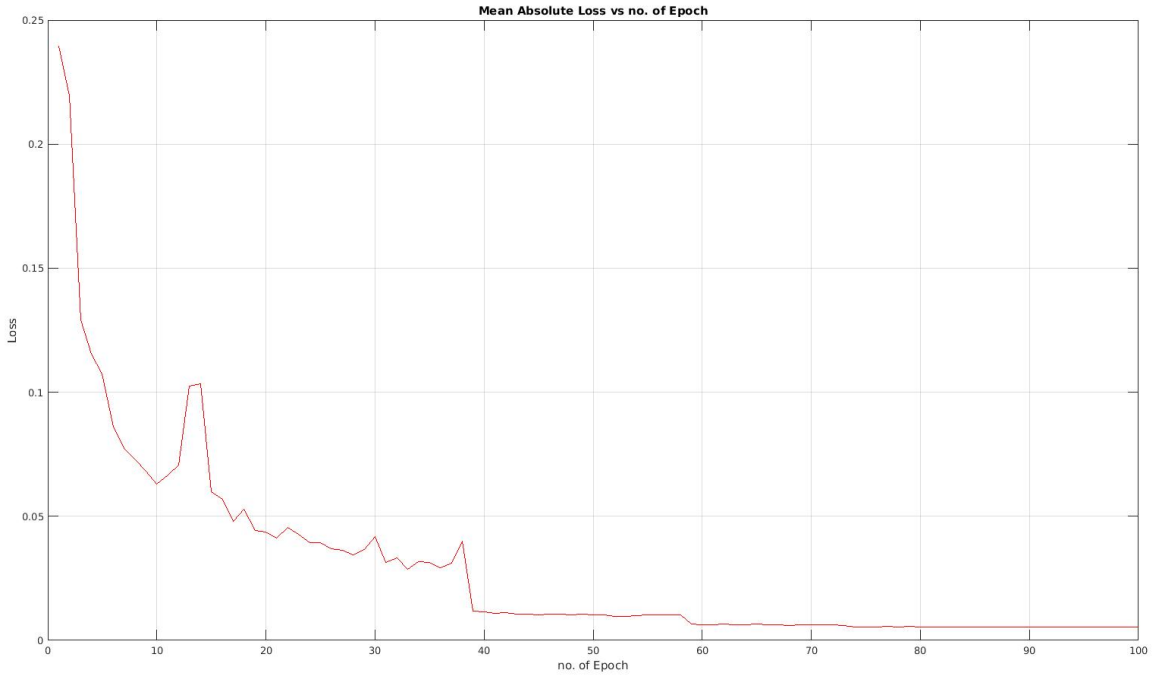


Figure 15: Trianing Loss vs No. of Epoch for MAE for Inception-V3

Convergence Graph for ResNet-50

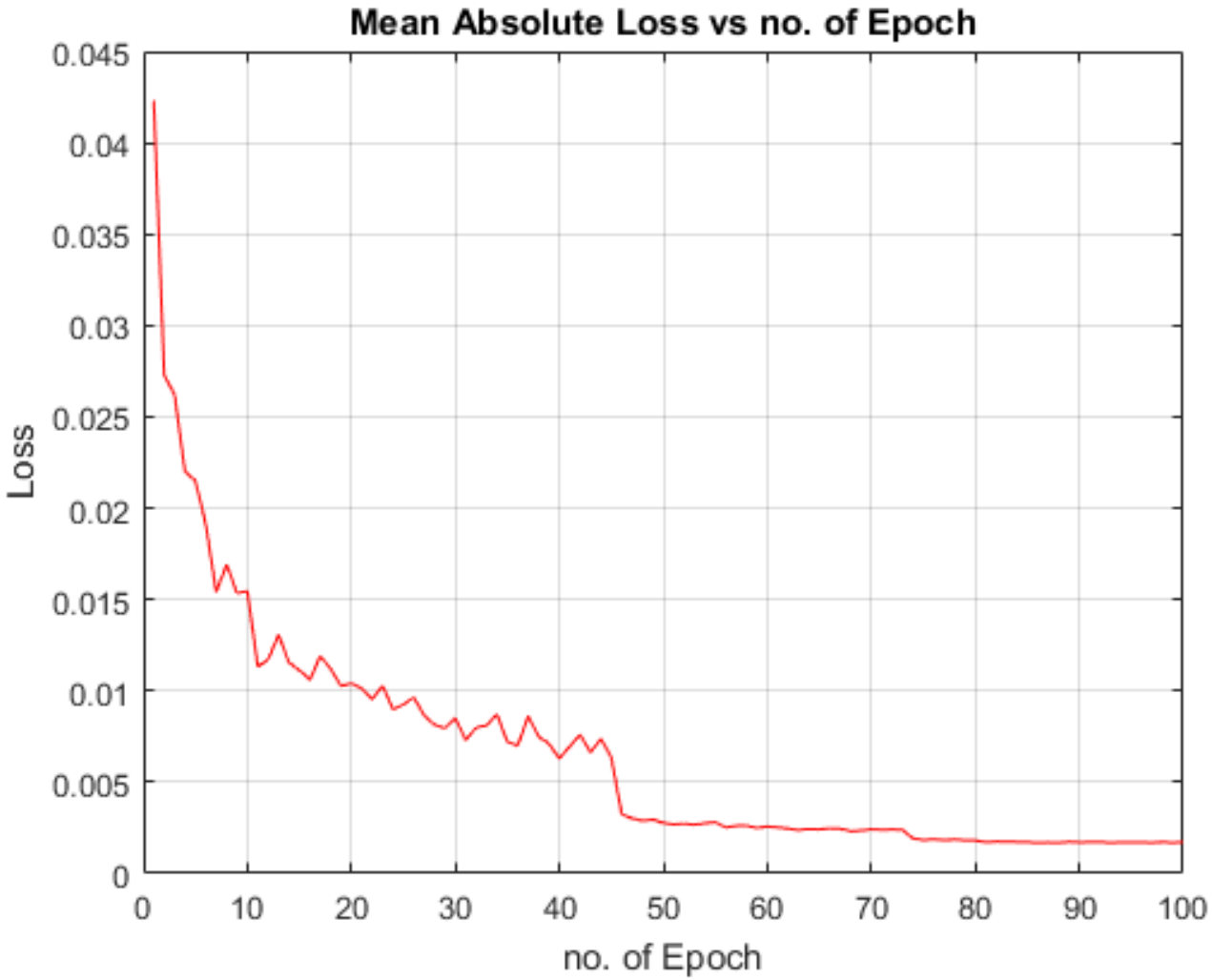


Figure 16: Trianing Loss vs No. of Epoch for MAE for ResNet-50

Convergence Graph for ResNet-101

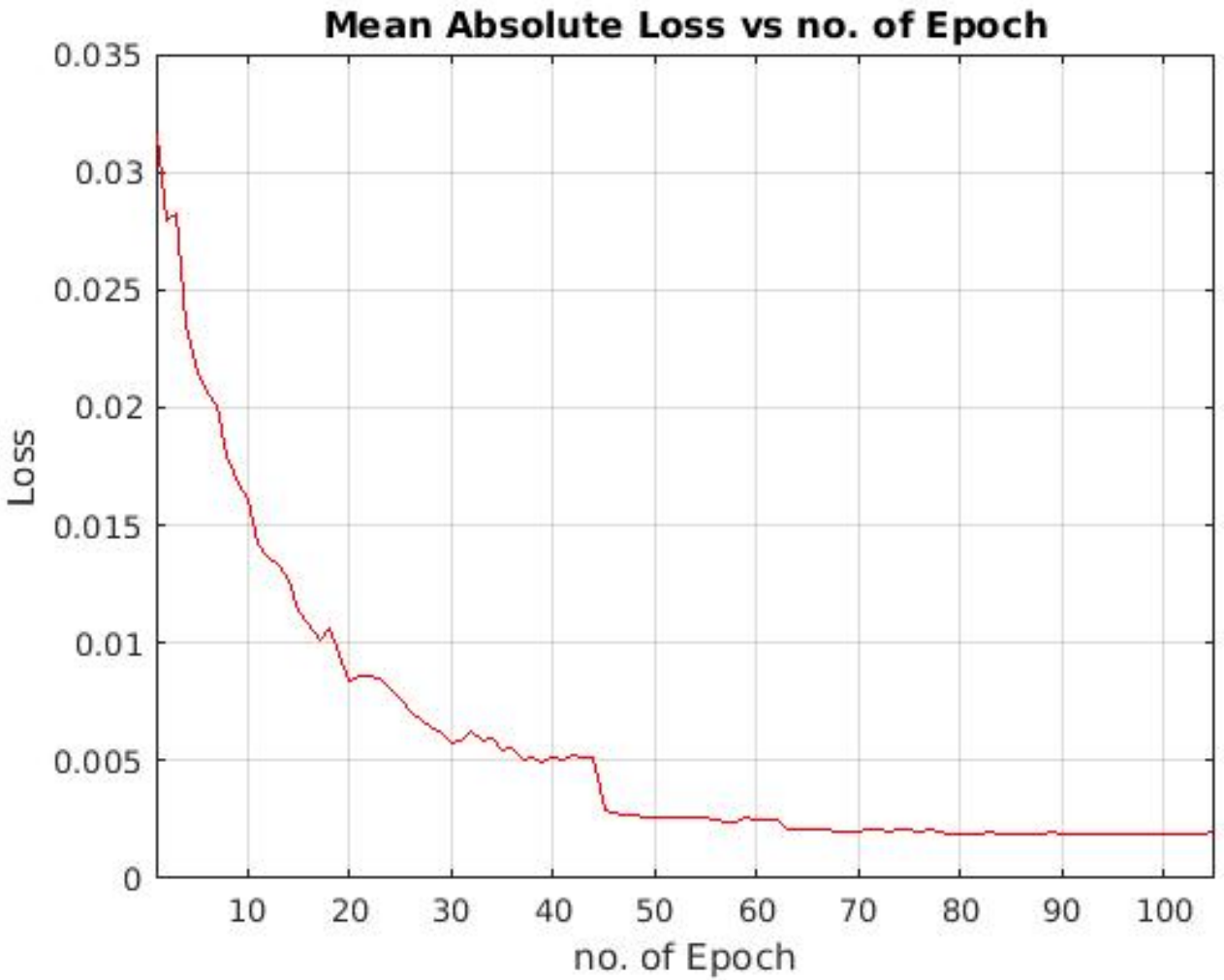


Figure 17: Trianing Loss vs No. of Epoch for MAE for ResNet-101

Convergence Graph for ResNet-201

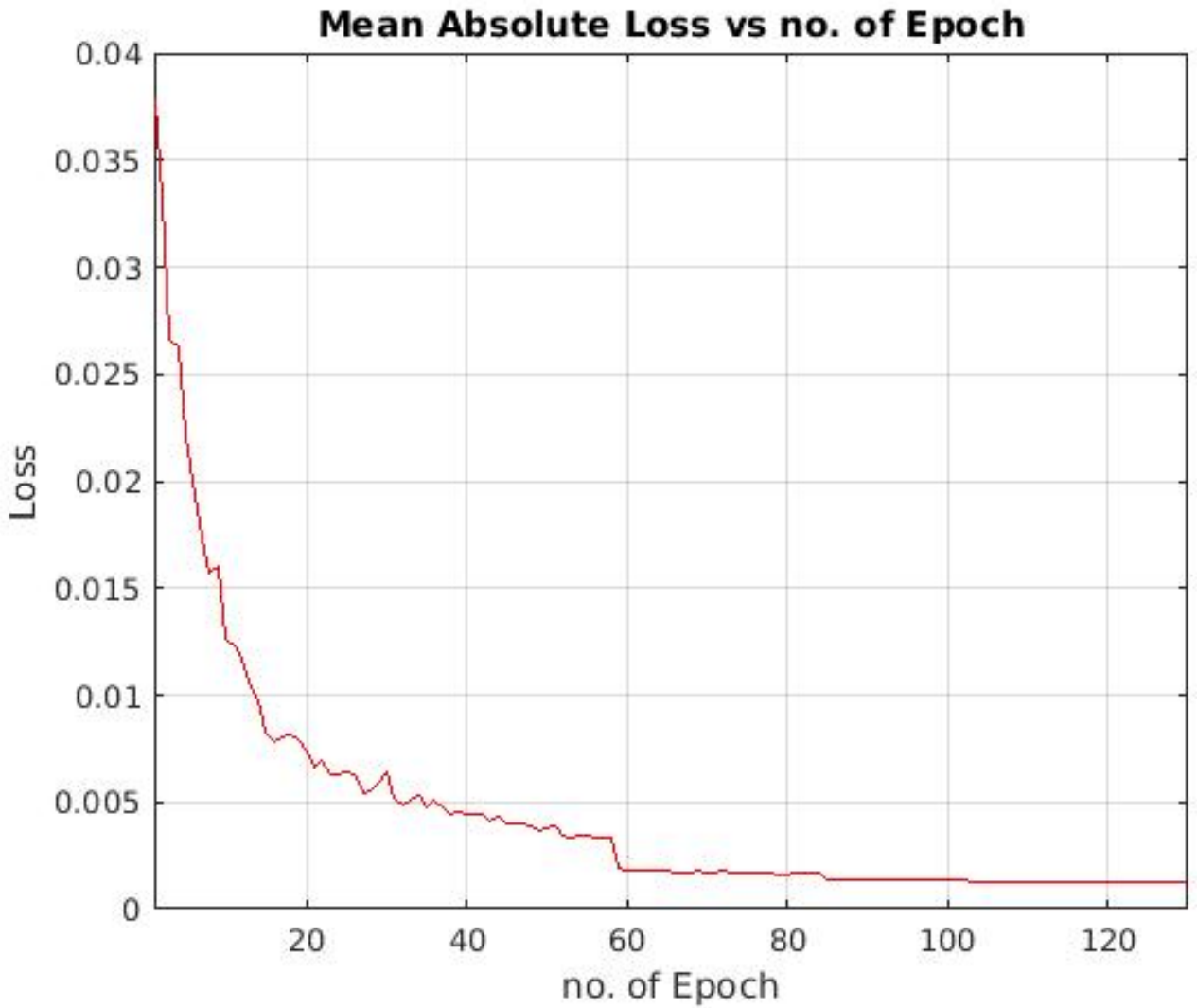


Figure 18: Trianing Loss vs No. of Epoch for MAE for ResNet-201

Testing Metrics

- We used three different metric to evalute the trained Model performance on test set, defined below

$$Mean\ Square\ Error(\hat{y}, y) = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2$$

$$Mean\ Absolute\ Error(\hat{y}, y) = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_t - y_t|$$

$$Mean\ Relative\ Error(\hat{y}, y) = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{y_t}$$

Here,
$\hat{y}$ is the prediction made on all the test examples
y is ground truth value of all the examples of test set.
$\hat{y}_t$ is the prediction made on t^{th} test example
y_t is ground truth value of t^{th} test set.
n is the total number of test example.

Performance over Test Set

Model Accuracy for Distance Prediction (in pixel)			
Model Name	**Mean Square Error**	**Mean Absolute Error**	**Mean Relative Error**
AlexNet	5.5677	27.0064	54.1467
DenseNet-161	0.0326	2.5060	1.557
DenseNet-201	0.0828	3.6442	12.1421
Inception-V3	0.0687	3.1364	10.4852
ResNet-50	0.0473	2.7485	9.0729
ResNet-101	0.1186	4.2163	14.6806
ResNet-201	0.06617	3.4258	10.8230

Table 1: Error in Distance Predicted

Distance Estimation by DenseNet-161 on Test Images

Model Performance When Drone tilted Right				
Images	**Ground Truth**	**Predicted Value**	**Abs. Diff**	**Location**
	**53.6**	**52.17**	**1.42**	CS-Dept
	**44**	**48.13**	**4.13**	TIIR

Result Cont...

Model Performance When Drone Facing Straight				
Images	Ground Truth	Predicted Value	Abs. Diff	Location
	174.93	169.01	5.91	Main Building
	156.2	157.84	1.581	TIIR

Result Cont...

Model Performance When Drone tilted Left				
Images	Ground Truth	Predicted Value	Abs. Diff	Location
	257.6	258.7	1.1725	Physics Dept
	262.39	266.31	3.912	TIIR

Conclusion

We summarize as follows :

- We performed **Pose Estimation**, one **Parameter** for **Localization** by modelling it to a **REGRESSION** task.
- This task is attained by distance calculated from the image captured from **Monocular Camera** with the help of pre-trained deep learning model.
- We have taken into consideration the Pilot perspective of ride, far novel than machine dominated navigation approach.

Future Work

We can formalize future work as follows:

- To propose a stopping condition i.e. Landing of the drone.
- To propose method to navigate in curvy corridor.
- To include more extreme condition and more building to enhance dataset .

References I

[1] Adam Bry, Abraham Bachrach, and Nicholas Roy.
State estimation for aggressive flight in gps-denied environments using onboard sensing.
In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al.
Imagenet large scale visual recognition challenge.
International Journal of Computer Vision, 115(3):211–252, 2015.

References II

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton.
Imagenet classification with deep convolutional neural networks.

In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[4] D. Lowe.
Distinctive image features from scale-invariant keypoints.
International Journal of Computer Vision, 60(2):91 – 110, February 2004.

[5] Ivan Edward Sutherland.
Sketchpad, a man-machine graphical communication system.
PhD thesis, Massachusetts Institute of Technology, January 1963.

References III

[6] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers.
Wallflower: Principles and practice of background maintenance.

In *Proceedings of International Conference on Computer Vision*, ICCV - 1999, pages 255 – 261, Kerkyra, Greece, September 1999.

[7] Eric Abbott and David Powell.
Land-vehicle navigation using gps.
Proceedings of the IEEE, 87(1):145–162, 1999.

[8] Jakob Engel, Thomas Schöps, and Daniel Cremers.
Lsd-slam: Large-scale direct monocular slam.
In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

References IV

[9] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus.
Regularization of neural networks using dropconnect.
In *International Conference on Machine Learning*, pages 1058–1066, 2013.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten.
Densely connected convolutional networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

References V

[12] Matthew D Zeiler and Rob Fergus.
Visualizing and understanding convolutional networks.
In *European conference on computer vision*, pages 818–833. Springer, 2014.

[13] Daniel Mellinger and Vijay Kumar.
Minimum snap trajectory generation and control for quadrotors.

In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[14] Karen Simonyan and Andrew Zisserman.
Very deep convolutional networks for large-scale image recognition.
arXiv preprint arXiv:1409.1556, 2014.

References VI

[15] Mark Müller, Sergei Lupashin, and Raffaello D'Andrea.
Quadrocopter ball juggling.
In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5113–5120. IEEE, 2011.

[16] Paul Checchin, Franck Gérossier, Christophe Blanc, Roland Chapuis, and Laurent Trassoudaine.
Radar scan matching slam using the fourier-mellin transform.
In *Field and Service Robotics*, pages 151–161. Springer, 2010.

[17] Christopher Mei, Gabe Sibley, Mark Cummins, Paul Newman, and Ian Reid.
Rslam: A system for large-scale mapping in constant-time using stereo.
International journal of computer vision, 94(2):198–214, 2011.

References VII

[18] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy.
Visual odometry and mapping for autonomous flight using an rgb-d camera.
In *Robotics Research*, pages 235–252. Springer, 2017.

[19] James F Roberts, Timothy Stirling, Jean-Christophe Zufferey, and Dario Floreano.
Quadrotor using minimal sensing for autonomous indoor flight.

In *European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, number LIS-CONF-2007-006, 2007.

References VIII

[20] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert.
Learning monocular reactive uav control in cluttered natural environments.
In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1765–1772. IEEE, 2013.

[21] Winston Davies and Pete Edwards.
Dagger: A new approach to combining multiple models learned from disjoint subsets.
machine Learning, 2000:1–16, 2000.

References IX

[22] Dean A Pomerleau.
Alvinn: An autonomous land vehicle in a neural network.
In *Advances in neural information processing systems*, pages 305–313, 1989.

[23] Markus Achtelik, Michael Achtelik, Stephan Weiss, and Roland Siegwart.
Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments.
In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 3056–3063. IEEE, 2011.

References X

[24] Michael Blösch, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart.
Vision based mav navigation in unknown and unstructured environments.
In *Robotics and automation (ICRA), 2010 IEEE international conference on*, pages 21–28. IEEE, 2010.

[25] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart.
Fusion of imu and vision for absolute scale estimation in monocular slam.
Journal of intelligent & robotic systems, 61(1-4):287–299, 2011.

References XI

[26] Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, and Nicholas Roy.
Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments.
In *Unmanned Systems Technology XI*, volume 7332, page 733219. International Society for Optics and Photonics, 2009.

[27] Cooper Bills, Joyce Chen, and Ashutosh Saxena.
Autonomous mav flight in indoor environments using single image perspective cues.
In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 5776–5783. IEEE, 2011.

[28] Abraham Bachrach, Ruijie He, and Nicholas Roy.
Autonomous flight in unknown indoor environments.
International Journal of Micro Air Vehicles, 1(4):217–228, 2009.

References XII

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al.
Going deeper with convolutions.
Cvpr, 2015.

[30] Niklas Karlsson, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E Munich.
The vslam algorithm for robust localization and mapping.
In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 24–29. IEEE, 2005.

[31] Dong Ki Kim and Tsuhan Chen.
Deep neural network for real-time autonomous indoor navigation.
arXiv preprint arXiv:1511.04668, 2015.

References XIII

[32] Ram Prasad Padhy, Feng Xia, Suman Kumar Choudhury, Pankaj Kumar Sa, and Sambit Bakshi.
Monocular vision aided autonomous uav navigation in indoor corridor environments.
IEEE Transactions on Sustainable Computing, 2018.

[33] Koray Çelik and Arun K Somani.
Monocular vision slam for indoor aerial vehicles.
Journal of electrical and computer engineering, 2013:4–1573, 2013.

Thank you!