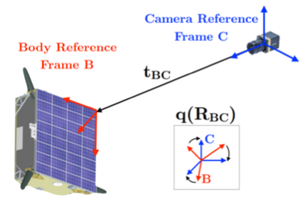
UC San Diego
JACOBS SCHOOL OF ENGINEERING

Group 23: Satellite Pose Estimation

ECE 228 Instructor: Peter Gerstoft



Arunkumar Ravichandran, Chinmayee Bhanu, Shehzeen Hussain, Sneha Kondur

{*arravich,cbhanu,ssh028,skondur*}@ucsd.edu

Motivation

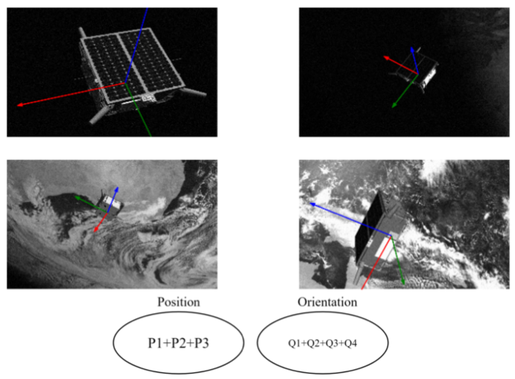
Uncooperative spacecraft servicing missions, automated removal of debris and autonomous docking of spacecrafts require accurate satellite pose estimation [1]

Problem Statement

- Estimate 3D pose of satellite object in uncluttered scene
- *Input*: Monochromatic image of the satellite in different backgrounds
- *Output*: Pose, 3D position of the object (x,y,z) in cartesian coordinates Orientation of the object represented using a Quaternion system, one real scalar and a three dimensional vector in complex space.
- Train a model on the synthetic image dataset and estimate on test image dataset

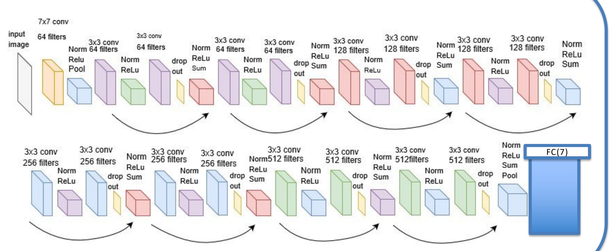
Dataset

- Dataset provided by Space Rendezvous Laboratory (SLAB) for their Satellite Pose Estimation Challenge on Kaggle.
- *Training set*: 12000 synthetic images of a satellite with corresponding ground truth pose labels as shown below
- *Test set*: 2998 synthetic image and 300 real images
- All images are 8 bit monochrome with a resolution 1920x1080



Methodology

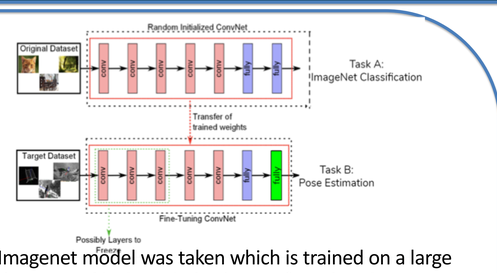
1. Preprocess the Dataset and apply suitable transformations
2. Architecture pretrained on the 1000 classes of ImageNet for classification task, with good generalization over other tasks.
3. Replace the last fully connected layer of 1000 neurons with 7 neurons to get real valued pose.



Improvements over Baseline

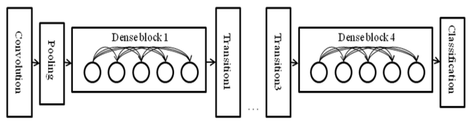
1. Resnet-50 is a deeper network (50 layers) than Resnet-18 (18 layers) so it extracts more informative features.
2. Freezing was done to reduce learning rate.

Model A: Resnet-50 Freezing Weights



1. Imagenet model was taken which is trained on a large dataset and its knowledge is transferred to the Resnet-50 with few layers frozen.
2. Imagenet has low-level features which will be learned by Resnet-50

Model B: Transfer Learning using Resnet-50



1. Since each layer in DenseNet receive all preceding layers as input, more diversified features can be learned.

Model C: Densenet-121

1. Motivation for a branched architecture stems from the sharing of weights for two complimentary but different tasks and usage of PoseLoss with it.

Model D: Posenet[2] inspired architecture

Loss Functions

The Predicted Pose Vector : $y^{\wedge} = [x^{\wedge}, q^{\wedge}]$
where,
$x^{\wedge}$ = *Predicted Position Vector*
$q^{\wedge}$ = *Predicted Orientation Vector*

The Ground Truth Pose Vector : $y = [x, q]$
where,
x = *Ground Truth Position Vector*
q = *Ground Truth Orientation Vector*

MSE Loss:
$$L = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - y^{\wedge(i)})^2$$

L1 Loss:
$$L = \sum_{i=1}^{n} | y^{(i)} - y^{\wedge(i)} |$$

PoseNet Loss:
$$L = \frac{1}{n}\sum_{i=1}^{n} ||x^{\wedge(i)} - x^{(i)}||_2 + \beta\, ||q^{\wedge(i)} - \frac{q^{(i)}}{|q^{(i)}|}||_2$$

$$\mathcal{L}_\beta(I) = \mathcal{L}_x(I) + \beta\mathcal{L}_q(I)$$

Weighted Loss

Results

Model	Real Image Error	Synthetic Image Error
Resnet-18 + MSE Loss(*Baseline*)	2.663	2.336
Resnet-50 Freezing (Model A)	2.569	2.140
Transfer Learning using Resnet-50 (Model B)	1.961	2.195
Densetnet-121 (Model C)	2.45	2.24
L1 Loss + Resnet-50	2.218	2.082
Weighted Loss + Resnet-50	2.59	1.74
Model D with PoseNet loss	1.859	1.722

**Lower error is better*

$$Error\ {}^{(i)}_{position} = \frac{|r^{(i)}_{gt} - r^{(i)}_{est}|_2}{|r^{(i)}_{gt}|_2}$$

$$Error\ {}^{(i)}_{orientation} = 2 \cdot \arccos\left(|\langle q^{(i)}_{est}, q^{(i)}_{gt}\rangle|\right)$$

$$Error\ {}^{(i)}_{pose} = Error\ {}^{(i)}_{orientation} + Error\ {}^{(i)}_{position}$$

N = number of images in test set

$$Error\ = \frac{1}{N}\sum_{i=1}^{N} Error\ {}^{(i)}_{pose}$$

Discussion

1. Since we had large number of training (12000) images, using Resnet-50 architecture provided significant improvements over Resnet-18
2. Densenet-121 didn't provide huge accuracy improvements because the dataset only has satellite images.
3. Training takes a long time ~1hour due to 12000 test images of higher resolution. Can be improved by using PoseCNN[2] weights and do transfer learning.
4. This model currently estimates pose on images but can be made to estimate pose dynamically in a video.
5. We found architecture changes only provided minimal accuracy improvement. But changing the loss function along with the architecture changes provided huge improvements in accuracy.

Future Work

1. Account for different backgrounds and clustered scenes.
2. Explore datasets with depth information in improving estimation accuracy.

References

[1] Sharma, Sumant, Connor Beierle, and Simone D'Amico. "*Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks*." 2018 IEEE Aerospace Conference. IEEE, 2018.

[2] Kendall, Alex, Matthew Grimes, and Roberto. "*Posenet: A convolutional network for real-time 6-dof camera relocalization*." Proceedings of the IEEE international conference on computer vision. 2015.

[3] Xiang, Yu, et al. "*Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes*." arXiv preprint arXiv:1711.00199 (2017).