Region-based Network for Yoga Pose Estimation with Discriminative Fine-Tuning Optimization

-International Journal of Intelligent Systems And Applications In Engineering, 25/07/2023

Authors:

- Shilpa Gite
- Deepak T. Mane
- Vijay Mane
- Sunil Kale
- Prashant Dhotre

Presentation By:

- Vivek Mange

Background

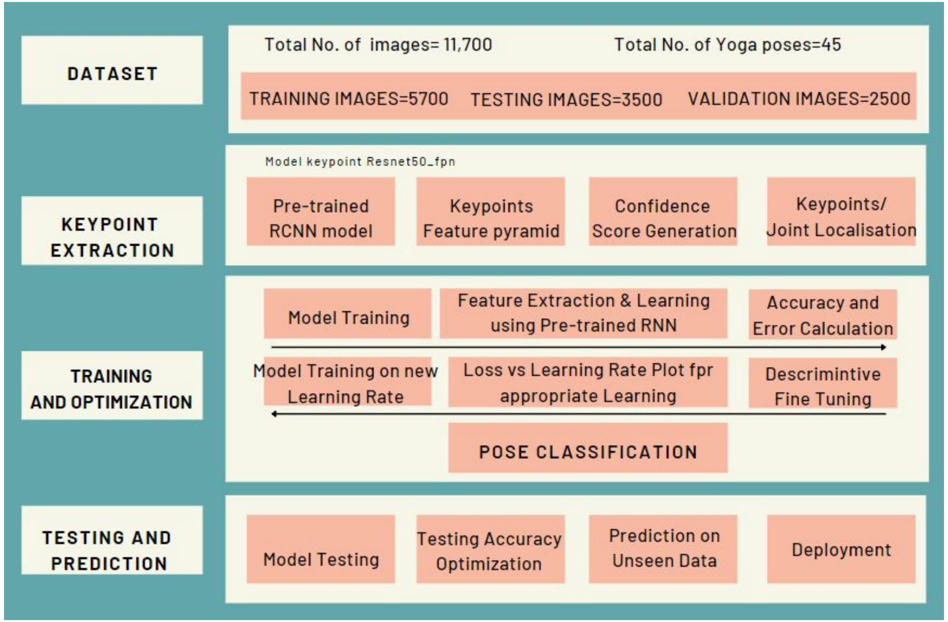
- Why do we need this?
  - Health Department - Therapy, Yoga classes, etc
  - Video Games or movies , Robotics - Rehab robots
- What is done here?
  - Implement ResNet and optimize, Dataset - Yoga-82, Accuracy - 90.5
- Initial challenges?
  - Huge no of variety of poses, Angles, lightings of images
  - Hidden key joints pose or overlapping poses
  - Distinguish target from background

Past Work on Yoga Pose

- **OpenPose** followed by **CNN and LSTM** (long short-term memory) hybrid model to get pose predictions. They achieved **99.38%** accuracy, but this model was only created for **six poses.**
- **BlazePose** is a **lightweight CNN** architectural model which analyzes **33 critical points** for pose estimation and is robust for real-world applications.
- **Challenges** - limitations of lighting, occluded images, changes in pose angles, Robust model including large number of complex yoga poses.

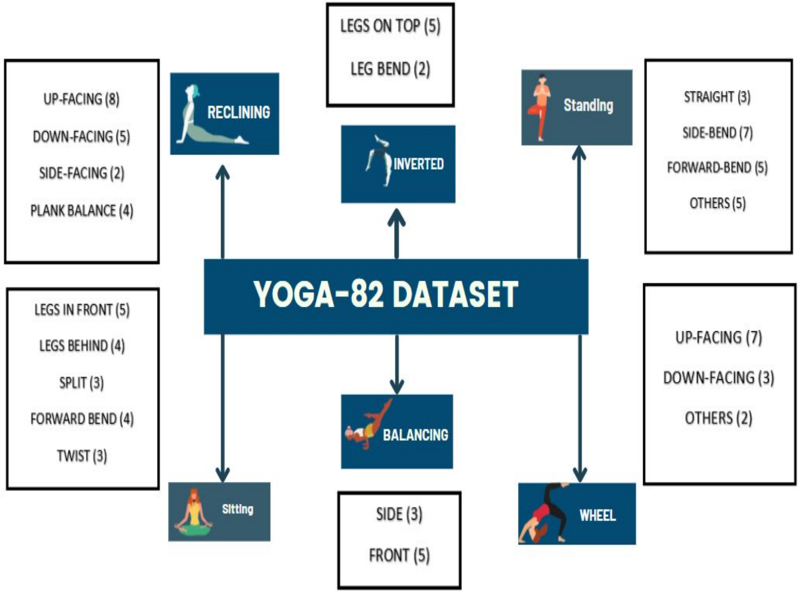
Method

- Pre-Possessing
- Key Point Detection and Skeleton Formation
- Pose Classification and Optimization
- Results



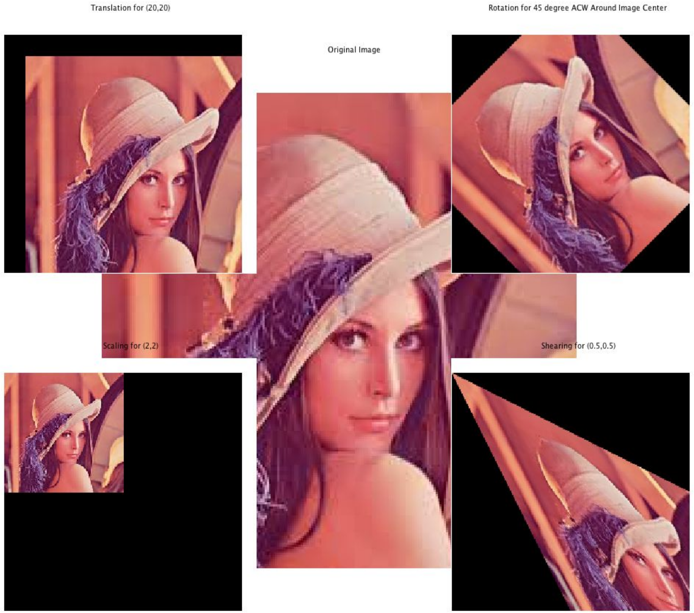
Dataset

- 28,000 yoga pose images, 82 yoga asanas with the hierarchical label
- used 45 classes for this research - 11,000 images spread across.
- Hierarchy
  - Standing(Stand,tree, etc)
  - Sitting (split, bend, etc)
  - Inverted (leg bend- scorpion, etc)
  - Wheel (cat-cow, etc)



Pre-Processing

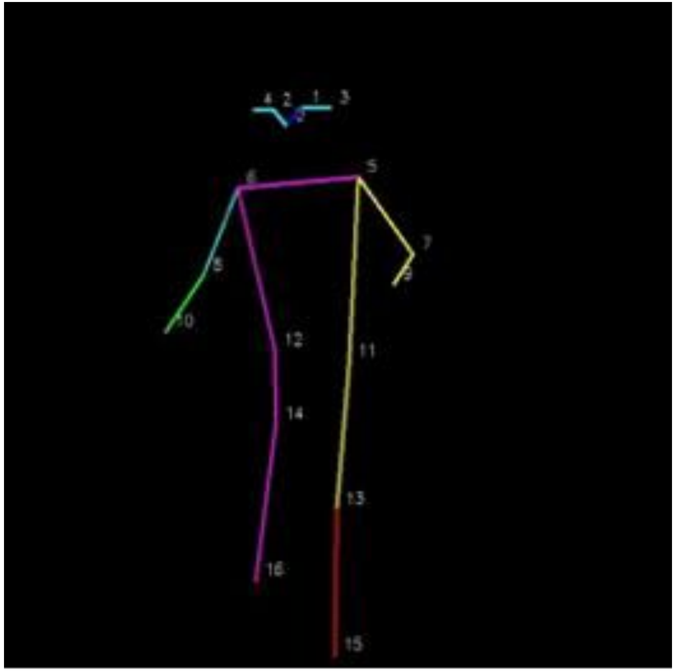
- Manual labeling
- Data Augmentation
  - rotated, skewed, sheared, zoomed, cropped, etc
  - Aim - reduce overfitting on the model
- Train and Test set
- Batch normalization for standardization
- Reshape and enhanced to fit model requirements



Image

Key Points

- Key-point-Resnet50_fpn (ResNet Feature Pyramid Network) network used for feature extraction
- Output is Detection Boxes, Confidence score and the key point
- 17 critical points Combined to form a skeleton. Structure of the pose combining;for (0, 1), (0, 2),etc

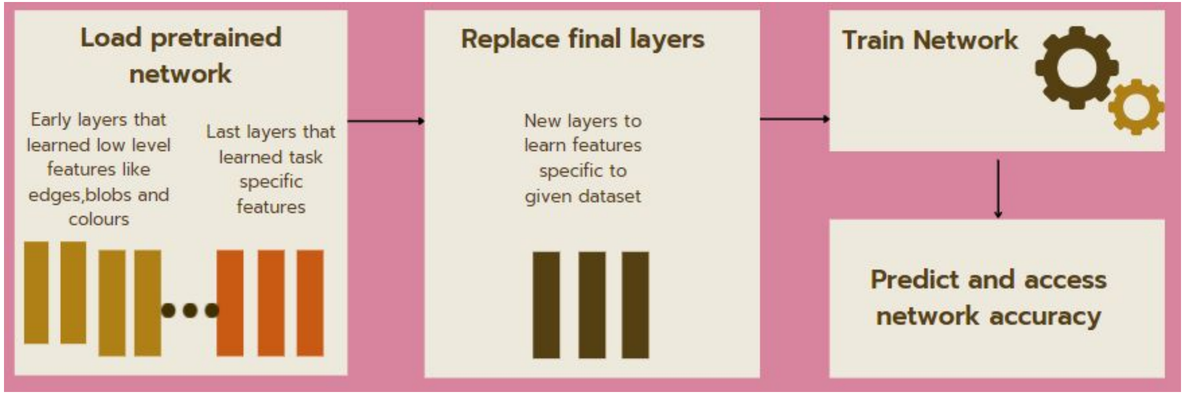


Key Points

Sr. No	Joint/Key-point	Sr. No	Joint/Key-point	Sr. No	Joint/Key-point
0	Nose	6	Right shoulder	12	Right hip
1	Left eye	7	Left elbow	13	Left knee
2	Right eye	8	Right elbow	14	Right knee
3	Left ear	9	Left wrist	15	Left ankle
4	Right ear	10	Right wrist	16	Right ankle
5	Left shoulder	11	Left hip		

Pose Classification

- Used Transfer Learning technique(CNN-Learner from Fastai)
- The detected key-points were trained
    - ResNet34
    - ResNet50
- ResNet34 has 34 deep layers
- ResNet50 has 48 deep layers along with 1 MaxPool and 1 Average Pool Layer



Initial Testing

ResNet50, Accuracy - 81.16%

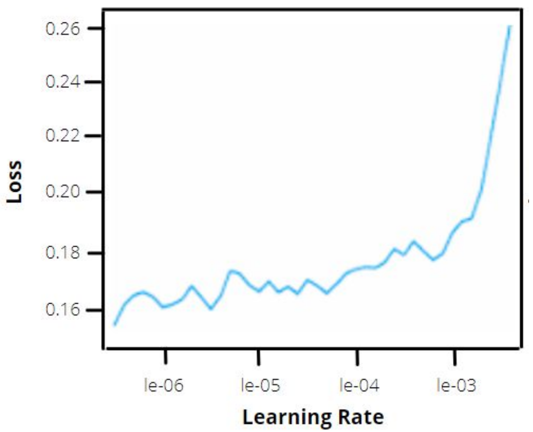
Epoch	Train_loss	Valid_loss	Accuracy	Time
0	3.5589	1.9186	0.4926	22:33
1	2.3018	1.4089	0.6137	05:26
2	1.6377	1.0957	0.6870	05:21
3	1.2089	0.9206	0.7392	05:24
4	0.9653	0.8229	0.7673	05:24
5	0.8139	0.7485	0.7896	05:26
6	0.6433	0.7146	0.7965	05:27
7	0.5471	0.6652	0.8093	05:25
8	0.4540	0.6515	0.8104	05:28
9	0.4083	0.6503	0.8116	05:28

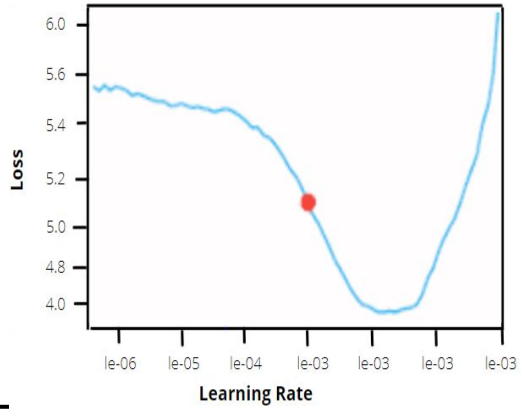
ResNet34, Accuracy - 78.61%

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	4.3934	2.4492	0.3726	06:22
1	2.6965	1.4727	0.5934	06:21
2	1.8465	1.2074	0.6635	06:31
3	1.4257	1.033	0.7064	06:36
4	1.1646	0.8977	0.7380	06:59
5	0.9618	0.8469	0.7554	07:00
6	0.8108	0.7924	0.7731	07:03
7	0.7311	0.7668	0.7838	07.08
8	0.6740	0.7579	0.7838	06:52
9	0.6239	0.7532	0.7861	06:52

Optimization

- Discriminative Fine-Tuning method
  - Training on all different layers of the network at different learning rates.
  - Focus on New layers
  - After optimizing
    - Learning rates:
      ResNet34 - 1e-06 to 1e-04
      ResNet50 - 1e-04 to 1e-02





Results after Optimization

ResNet50 - 90.5%

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	0.7676	1.4320	0.6780	14:00
1	1.3870	2.2410	0.5482	05:31
2	1.3047	1.1580	0.6689	05:33
3	0.9908	0.9102	0.9102	05:31
4	0.7304	0.6251	0.8333	05:36
5	0.5330	0.4585	0.8704	05:36
6	0.3362	0.3748	0.8962	05:37
7	0.2352	0.3568	0.9055	05:30

ResNet34 - 81.57%

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	0.6919	0.7425	0.7823	12:39
1	0.5991	0.7072	0.7936	05:31
2	0.5614	0.6726	0.7988	05:31
3	0.5100	0.6496	0.8102	05:31
4	0.4527	0.6329	0.8122	05:35
5	0.4206	0.6253	0.8157	05:39
6	0.4019	0.6192	0.8168	05:37
7	0.4197	0.6230	0.8157	05:37

Final output images

Cat_Cow_Pose_or_Marjaryasana_

Eight_Angle_Pose_or_Astavakrasana

Final output images

Gate_Pose_or_Parighasana

Plank_Pose_or_Kumbhaksana

Comparison

- **OpenPose** architecture with CNN and LSTM - 99% accuracy, But has **more false positives** on animals and statues and **struggle in overlapping poses.**
- **PoseNet**: **Poor performance** in **horizontal poses** like Balancing poses.
- **MR-CNN**: The network is **slow** to decline during the training weight parameters may **fail** to find the global **optimal solution.**

Method	Dataset	Accuracy
MR-CNN	MS COCO, PASCAL, VOC	89.3%
CNN-LSTM	6 Poses, 12 People	98.92%
BLAZEPOSE	1000 Pictures	97.2%
OPENPOSE	AR Dataset	87.8%
SVM	6 Poses, 15 People	98.58%
ResNe34	Yoga-82 dataset (11,000 images with 45 classes)	81.5%
ResNet50		90.5%

Conclusion

- The proposed method extracts the essential 17 key points (body joints) from an image
- forms a skeletal structure to examine the posture
- key points are trained by the ResNet50 model, which acts as a pose classification model
- The result gives an accuracy of 90.5% over 45 different classes.