The Realtime 2D Pose Estimation

Project Report submitted to

Rajiv Gandhi University of Knowledge Technologies, Basar

for the partial fulfilment of the requirements

for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

M.Preethi - ID No: B141475

CH.Srujana - ID No: B141755

R.Sandhya rani- ID No: B141349

Under the guidance of

Ms.U Nagamani

Assistant Professor in the Dept. Of CSE

Department of Computer Science & Engineering

Rajiv Gandhi University of Knowledge Technologies

Basar, Nirmal(Dist), Telangana – 504107

March, 2020

THE REALTIME 2D POSE ESTIMATION

M.Preethi
ID:B141475

CH.Srujana
ID:B141755

R.Sandhya rani
ID:B141349

Under the guidance of

Mrs.U Nagamani

Assistant Professor in the Dept. Of CSE

Department of Computer Science & Engineering

Rajiv Gandhi University of Knowledge Technologies

Basar, Nirmal(Dist), Telangana – 504107

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE

TECHNOLOGIES, BASAR

CERTIFICATE

This is to certify that the Project Report entitled "**Realtime 2D Pose Estimation**", submitted by **M.Preethi ID: B141475, CH.Srujana ID:B141755, R.Sandhya rani ID:B141349**, Department of Computer Science and Engineering, Rajiv Gandhi University of Knowledge Technologies, Basar, for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering is a bonafied record of the work and investigations carried out by him/her under my supervision and guidance.

Project Supervisor
Mrs.U Nagamani
Assistant Professor

Head of the Department
Mrs.G.Srujana
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,

BASAR

DECLARATION

We hereby declare that the work which is being presented in this project entitled, "Real Time Multi-person 2D Pose Estimation using Part Affinity Fields" submitted to RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, BASAR in the partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOG in COMPUTER SCIENCE AND ENGINEERING, is an authentic record of our own work carried out under the supervision of Mrs.U Nagamani, Assistant Professor in Department of Computer Science and Engineering, RGUKT, Basar. The matter embodied in this project report has not been submitted by us for the award of any other degree

M.Preethi(B141475)

CH.Srujana(B141755)

R.Sandhya rani(B141349)

Place: Basar

Date: 17-04-2020

ACKNOWLEDGEMENT

We take this opportunity to express our deep and sincere gratitude to our Supervisor Ms. U. Nagamani for her valuable guidance and for giving us the opportunity to work under her. Her constantencouragement, support and generous attitude were a tremendous boost for our work. We would like to express our sincere gratitude to our HOD Ms. Srujana for encouraging us. We also express our sincere thanks to our Project Coordinator Mr.Laxminarayana for providing guidance during the evaluation processes. We thank all our seniors, friends, and members of our project group in materializing this report and also for the lively support and encouragement given from time to time. We have great regard, and we wish to extend our warmest thanks to our classmates who offered constant support during one year span of dissertation work.

M.Preethi

CH.Srujana

R.Sandhaya rani

ABSTRACT

We present an approach to efficiently detect the 2D pose of multiple people in an image. The approach uses a non-parametric representation, which we refer to as Part Affinity Fields(PAFs), to learn to associate body parts with individuals in the image. The architecture encodes global context, allowing a greedy bottom-up parsing step that maintains high accuracy while achieving realtime performance, irrespective of the number of people in the image. The architecture is designed to jointly learn part locations and their association via two branches of the same sequential prediction process. We have used the method called COCO (**Common Objects in Context**) keypoints.

Keywords: PAFs, COCO

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 Motivation

Human pose estimation is widely used in humancomputer interactions, gaming, virtual reality, video surveillance, sports analysis, and medical assistance,making it a highly popular research topic in the field of computer vision. Human pose estimation aims to automatically locate the human body parts from images or videos. Where only one person is in the image or the position of the person is given, a single-person algorithm should be performed to locate the human parts, such as the top of the head, the center of the neck, the left/right elbows,and the left/right shoulders

In more general cases,where the number and the position of persons in an image are unknown, the multi-person pose estimation algorithms are performed. All the human parts in an image should be detected and the keypoints of the same person, even in a crowded scene, should be associated.

Fig: Multi-person pose estimation

1.2 Problem Definition

Human 2D pose estimation the problem of localizing anatomical keypoints or “parts” has largely focused on finding body parts of individuals. Inferring the pose of multiple people in images, especially socially engaged individuals, presents a unique set of challenges. First, each image may contain an unknown number of people that can occur at any position or scale. Second, interactions between people induce complex spatial interference, due to contact, occlusion, and limb articulations, making association of parts difficult. Third, runtime complexity tends to grow with the number of people in the image, making realtime performance a challenge.

A common approach is to employ a person detector and perform single-person pose estimation for each detection. These top-down approaches directly leverage existing techniques for single-person pose estimation, but suffer from early commitment: if the person detector fails as it is prone to do when people are in close proximity there is no recourse to recovery. Furthermore, the runtime of these top-down approaches is proportional to the number of people for each detection, a single-person pose estimator is run, and the more people there are, the greater the computational cost. In contrast, bottom-up approaches are attractive as they offer robustness to early commitment and have the potential to decouple runtime complexity from the number of people in the image. Yet, bottom-up approaches do not directly use global contextual cues from other body parts and other people. In practice, previous bottom-up methods do not retain the gains in efficiency as the final parse requires costly global inference.

1.3 Organization of the Report

This report is divided into the following chapters

Chapter 2:Literature Survey- Here description of what techniques and algorithms exist to solve the mentioned problem is given.

Chapter 3:Proposed Method- In this section, the proposed algorithms and techniques have been given.

Chapter 4:Implementation of Algorithm- In this section, the implementation details of algorithm are given.

Chapter 5:Test and Results- This section, describes the test plan and Test cases along with the outputs and references.

Chapter 6:Conclusion- Finally we conclude that algorithm can solve search problem with less time complexity.

CHAPTER 2

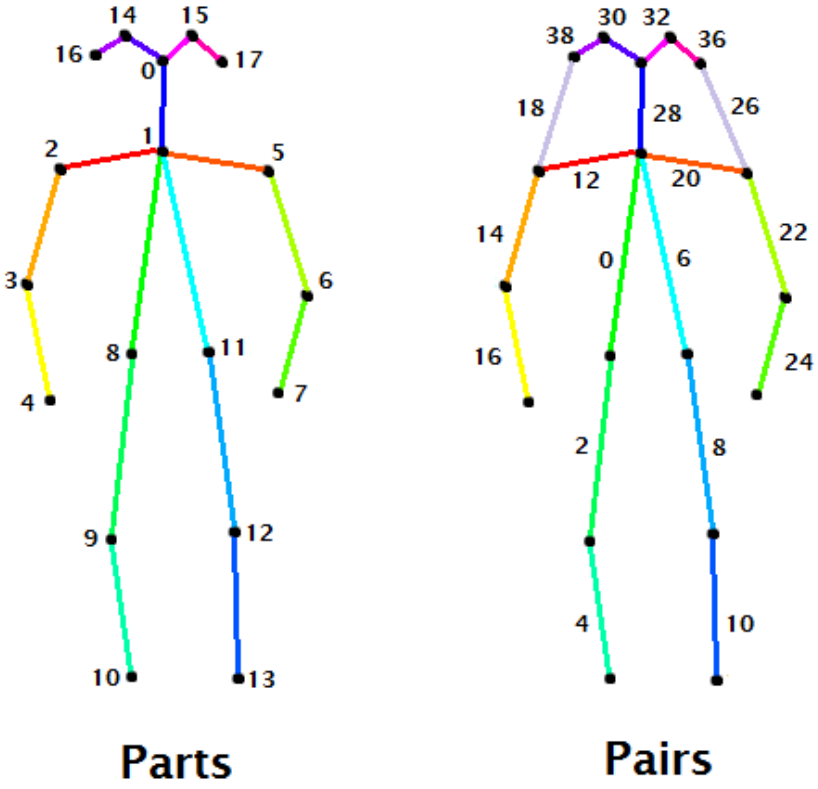
LITERATURE SERVEY

2.1 Pose Estimation

Pose Estimation is a general problem in Computer Vision where we detect the position and orientation of an object. This usually means detecting keypoint locations that describe the object.In this we will focus on human pose estimation, where it is required to detect and localize the major parts/joints of the body (e.g. shoulders, ankle, knee, wrist etc.).

An approach to efficiently detect the 2D pose of multiple people in an image. The approach uses a non-parametric representation, which we refer to as Part Affinity Fields (PAFs), to learn to associate body parts with individuals in the image.

A Human Pose Skeleton represents the orientation of a person in a graphical format. Essentially, it is a set of coordinates that can be connected to describe the pose of the person. Each co-ordinate in the skeleton is known as a part (or a joint, or a keypoint). A valid connection between two parts is known as a pair (or a limb).Note that, not all part combinations give rise to valid pairs. A sample human pose skeleton is shown below.



A body **part** is an element of the body, like neck, left shoulder or right hip.

A **pair** is a couple of parts. A *connection* between parts. We could say limb, but the connection between the nose and the left eye is definitely not a limb. Also, we are going to deal with connections between ears and shoulders that do not exist in real life.

2.2 Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

2.3 Approaches

2.3.1 Top Down approach

The simple approach is to incorporate a person detector first, followed by estimating the parts and then calculating the pose for each person. This method is known as the **top-down** approach.The top-down approach starts by identifying and localizing individual person instances using a bounding box object detector. This is then followed by estimating the pose of a single person.

Top down approaches involve a segmentation step at the start, where each human is first segmented into a bounding box, followed by pose estimation being performed individually on each bounding box. Top down pose estimation can be classified into generative body-model based and deep learning based approaches. Generative body-model based approach involves trying to fit a body-model on the image, allowing the final prediction to be human like. Deep Learning based approached directly predict joint locations, thus the final prediction have no guarantee of being human like.

2.3.2 Bottom Up approach

Another approach is to detect all parts in the image (i.e. parts of every person), followed by associating/grouping parts belonging to distinct persons. This method is known as the bottom-up approach.The bottom-up approach starts by localizing identity-free semantic entities, then grouping them into person instances.

Bottom up approaches involve first detecting the parts or joints for one or more humans in the image, and then assembling the parts together and associating them with a particular human. In simpler terms, the algorithm first predicts all body parts/joints present in the image. This is typically followed by the formulation of a graph, based on the body model, which connects joints belonging to the same human. Integer linear programming (ILP) or bipartite matching are two common methods of creating this graph.

2.4 Technologies Used

Python Programming

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power.

Tensorflow

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or *tensor*.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

TensorFlow applications can be run on most any target that's convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google's own cloud, you can run TensorFlow on Google's custom TensorFlow Processing Unit (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

TensorFlow benefits

The single biggest benefit TensorFlow provides for machine learning development is *abstraction.* Instead of dealing with the nitty-gritty details of implementing algorithms, or figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application. TensorFlow takes care of the details behind the scenes.

TensorFlow offers additional conveniences for developers who need to debug and gain introspection into TensorFlow apps. The eager execution mode lets you evaluate and modify each graph operation separately and transparently, instead of constructing the entire graph as a single opaque object and evaluating it all at once. The TensorBoard visualization suite lets you inspect and profile the way graphs run by way of an interactive, web-based dashboard.

TensorFlow also gains many advantages from the backing of an A-list commercial outfit in Google. Google has not only fueled the rapid pace of development behind the project, but created many significant offerings around TensorFlow that make it easier to deploy and easier to use: the above-mentioned TPU silicon for accelerated performance in Google's cloud; an online hub for sharing models created with the framework; in-browser and mobile-friendly incarnations of the framework and much more.

OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library which is generally used for image processing. It is first developed in Nizhny Novgorod where the Intel's research center is located by Intel researchers.Now it is supported by Itseez. Its library has more than 2500 optimized algorithms, which include both old and new computer vision and machine learning algorithms. These algorithms are used to detect faces, objects, sorting human reactions in video frames, follow eye movements, and track any detected object. It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.In this project OpenCV libraries is used for object detection on real time video frames.

OpenCV Benefits

Cross-Platform Library

OpenCV is a cross-platform library, enabling users on different operating systems to use it without complications. It is even accessible on mobile systems like iOS and Android, making it a truly portable library.

Vast Algorithms

OpenCV offers users access to over 2,500 algorithms, both classic and state-of-the-art. This vast library allows programmers to perform a multitude of tasks in their software such as extracting 3D models of objects, removing red eyes, following eye movements, and more.

Extensive Use

OpenCV is being used by giant companies like Google, IBM, Toyota and startups such as Applied Minds and Zeitera as well as organizations in countries all over the world to conduct multifarious tasks. This gives users assurance that they are onboarding a library that is being used extensively by enterprises and government institutions.

Moreover, OpenCV has a vast community where users can ask for assistance and offer help to their fellow developers in case they have questions regarding codes or the platform. This lets developers access insights from real people about the library and its codes.

CHAPTER 3

3.1 Existing System

Single-person approaches estimate human pose in an image or a video under the condition that the position of the human is given. Generally, the position and rough scale of a person or the bounding box of a person are provided before estimation. Early works model human parts as a stickman but recent works model it as key joints because such joints are connected naturally and have more accurate positions. The objective of deep learning based single-person approaches is to locate keypoints of human parts. This one directly regresses keypoints from the features, and we call it direct regression based framework.

Direct regression based framework

Several works are based on direct regression framework. Toshev and Szegedy [5] proposed a cascaded DNN regressor to predict human keypoints directly. However, it is difficult to learn mapping directly from feature maps without other procedures. By feeding back error predictions, the predicted keypoint locations are refined progressively. Proposed a structure-aware approach called “compositional pose regression”. Unlike other related works, this approach re-parameterizes pose representation using bones instead of joints, which is more primitive, stable, and easier to learn. Long-range interactions between bones are encoded by a compositional loss function. proposed Soft-argmax to convert heatmaps to coordinates in a fully differentiable fashion. A keypoint error distance based loss function and a context-based structure are utilized in their end to end trainable network, enabling it to achieve comparable results to the state-of-the-art heatmap based framework.

Earlier works and a few recent works have attempted to regress the coordinates of keypoints directly. The direct regression of joint positions is highly non-linear and has difficulty in learning mapping. Furthermore, it cannot be applied to a multi-person case (bottom-up approaches or a single detection box containing more than one person).When direct regression is applied, the final result can be obtained in an end-to-end fashion without handling heatmaps.

3.1 Proposed System

Compared with the single-person pipeline, the multi- person pipeline is more difficult because neither the number nor the position of the person is given. Keypoint detection and human location are two core problems in this task. To solve these two problems, two popular pipelines have been proposed: (1) top-down pipeline and (2) bottom-up pipeline.

Preprocessing

First but not least, we convert the image from [0, 255] to [-1, 1]. Though it's the easiest step of all, I failed miserably on this in my first post of these series.

Neural Network

It's the same dark force that makes planes fly and smartphones exist. We'd better treat it like a black box. The last operation of the neural network returns a tensor consisting of 57 matrices. However, this last op is just a concatenation of two different tensors: heatmaps and PAFs. The understanding of these two tensors is essential.

Heatmap

A heatmap is a matrix that stores the confidence the network has that a certain pixel contains a certain part. There are 18 (+1) heatmaps associated with each one of the parts and indexed. We will extract the **location of the body parts** out of these 18 matrices.

PAFs

Part Affinity Fields are matrices that give information about the position and orientation of pairs. They come in couples: for each part we have a PAF in the 'x' direction and a PAF in the 'y' direction. There are 38 PAFs associated with each one of the pairs and indexed. We will **associate couples of parts into pairs** thanks to these 38 matrices.

Merging

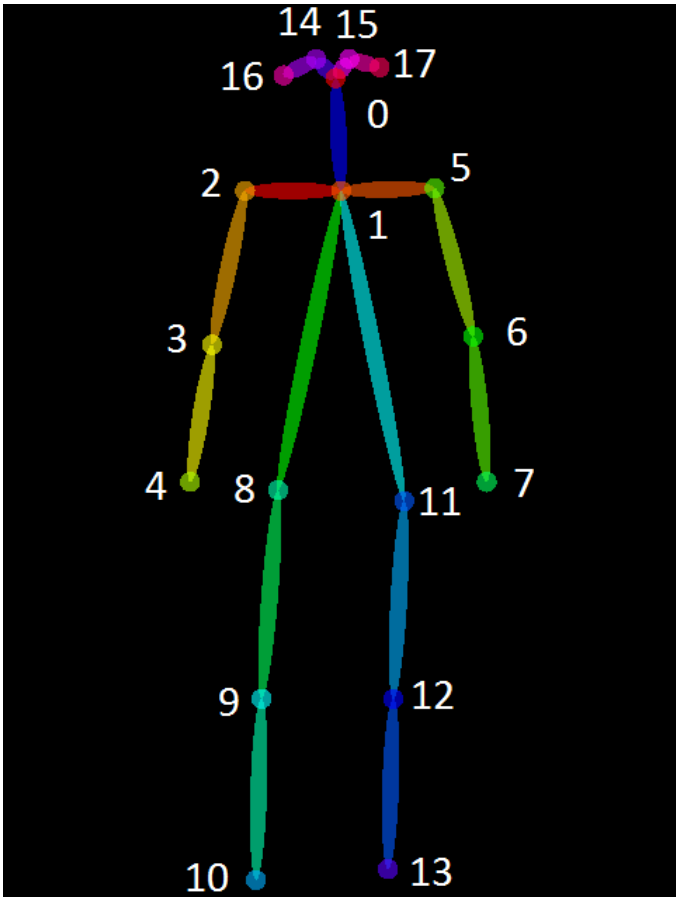
The final step is to transform these detected connections into the final skeletons.

CHAPTER 4

4.1 Dataset

COCO stands for Common Objects in Context. As hinted by the name, images in COCO dataset are taken from everyday scenes thus attaching “context” to the objects captured in the scenes. We can put an analogy to explain this further. Let’s say we want to detect a person object in an image. A non-contextual, isolated image will be a close-up photograph of a person. Looking at the photograph, we can only tell that it is an image of a person. However, it will be challenging to describe the environment where the photograph was taken without having other supplementary images that capture not only the person but also the studio or surrounding scene.

COCO was an initiative to collect natural images, the images that reflect everyday scene and provides contextual information. In everyday scene, multiple objects can be found in the same image and each should be labeled as a different object and segmented properly. COCO dataset provides the labeling and segmentation of the objects in the images. A machine learning practitioner can take advantage of the labeled and segmented images to create a better performing object detection model.The COCO(Common Objects in Context) train, validation, and test sets, containing more than 200,000 images and 250,000 person instances labeled with keypoints are available and COCO model produces 18 keypoints.we will download the model which is trained on the COCO Dataset.

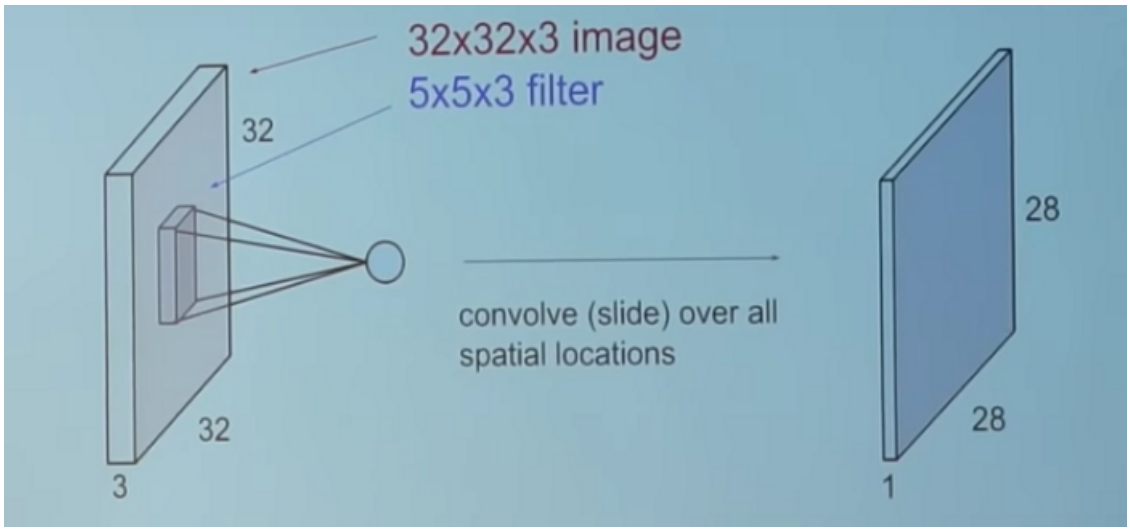


4.2 Convolutional neural network

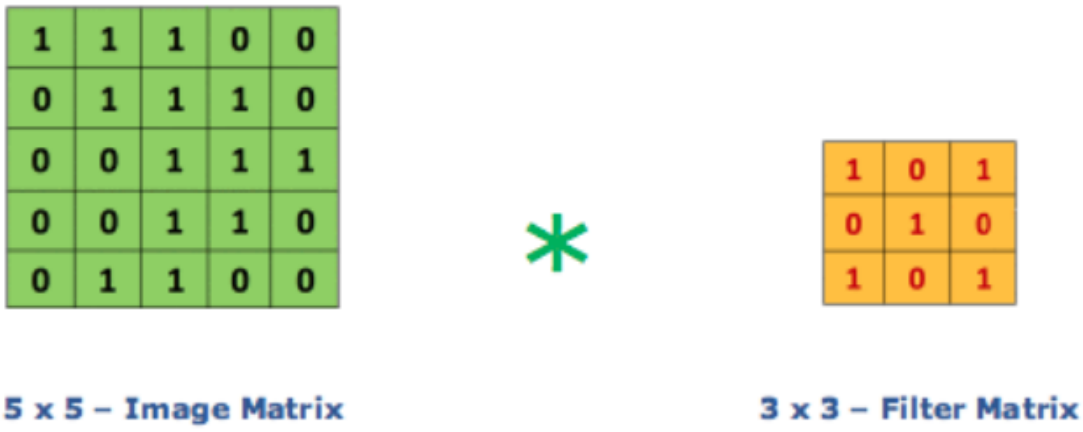
A convolutional neural network is used for our model. A convolutional neural network is similar to an ordinary neural network and contains neurons and weights for each neuron. While a regular neural net doesn’t scale well to take full images as input,a convolutional neural net can take large images as input and their architecture is designed accordingly. Three main layers areused for a convolutional neural net: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. The initial layers of theneural net are used for feature extraction while the fully connected layers predict the coordinates and output probabilities. Thisnetwork has 24 initial convolutional layers and 2 fully connected layers.Finally, an input image (resized to 416x416 pixels) is passed to the convolutional neural net in a single pass, which comes out as a7x7x30 tensor, describing the bounding boxes for grid cells. The final scores for the bounding boxes are calculated and the ones having low scores are discarded.

What is Convolution?

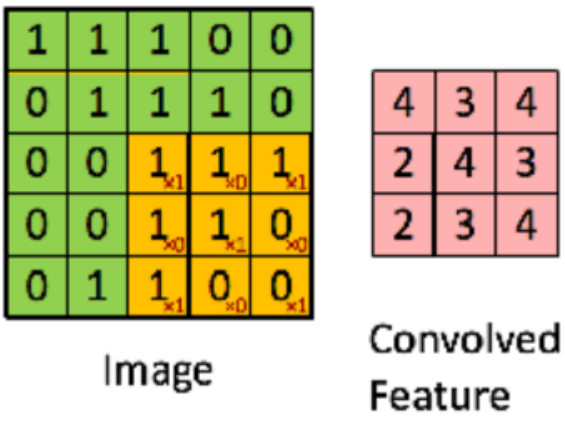
Convolution is the first layer to extract features from an input image. It preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.



From the above image we can observe that for our input of 32*32*3 we took a filter of 5*5*3 and slided it over the complete image and along the way take the dot product between the filter and chunks of the input image. The output results with an image of size 28*28*1.



Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called "Feature Map" as output shown in below with a stride of 1.



Stride: Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.

Padding: If we increase the stride value the size of image keeps on reducing, padding with zeros across it solves this problem.From the below image we can observe the size of image is retained after padding zeros with stride 1.

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

WEIGHT

1	0	1
0	1	0
1	0	1

139	184	250	409	410	283
133	429	505	686	856	441
310	261	792	412	640	341
280	633	653	851	751	317
254	608	913	713	657	503
321	325	592	517	637	78

ReLU Activation: ReLU stands for Rectified Linear Unit for a non-linear operation. Its purpose is to introduce non-linearity which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

Pooling: Sometimes when the images are too large, we would need to reduce the number of trainable parameters. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is done for the sole purpose of reducing the spatial size of the image. Pooling is done independently on each depth dimension, therefore the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling.

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

792	856
913	851

Here we have taken stride as 2, while pooling size also as 2. The max operation is applied to each depth dimension of the convolved output. As you can see, the 4*4 convolved output has become 2*2 after the max pooling operation.

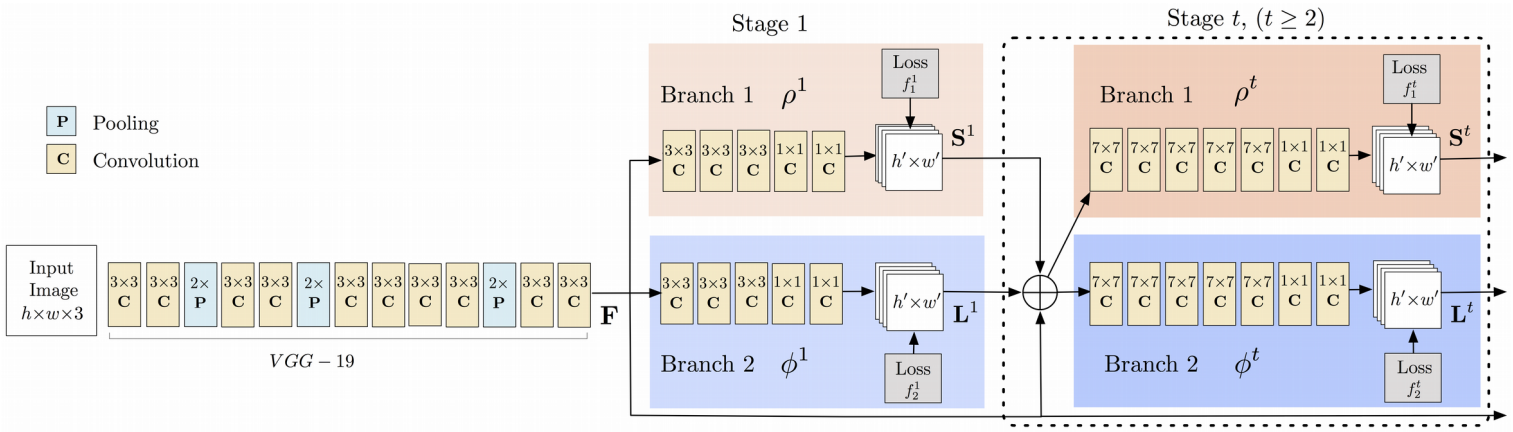


Fig: Multi-Person Pose Estimation model architecture

The model takes as input a color image of size w × h and produces, as output, the 2D locations of keypoints for each person in the image. The detection takes place in three stages :

Stage 0: The first 10 layers of the VGGNet are used to create feature maps(F) for the input image.

Stage 1: A 2-branch multi-stage CNN is used where the first branch predicts a set of 2D confidence maps (S) of body part locations (e.g. elbow, knee etc.). Given below are confidence maps and Affinity maps for the keypoint – Left Shoulder.

The second branch predicts a set of 2D vector fields (L) of part affinities, which encode the degree of association between parts. In the figure below part affinity between the Neck and Left shoulder is shown.At the first stage, the network produces a set of detection confidence maps S1 = ρ1(F) and a set of part affinity fields L1 = φ1 (F), where ρ1 and φ1 are the CNNs for inference at Stage1. In each subsequent stage, the predictions from both branches in the previous stage, along with the original image features F, are concatenated and used to produce refined predictions,

$$S^t = \rho^t(F, S^{t-1}, L^{t-1}) , \quad \forall t \geq 2$$

$$L^t = \varphi^t(F, S^{t-1}, L^{t-1}) , \quad \forall t \geq 2$$

CHAPTER 5

5.1 Results

We evaluate our method on coco dataset challenge for multi-person pose estimation. The COCO keypoints challenge dataset collect images in diverse scenarios that contain many real-world challenges such as crowding, scale variation, occlusion, and contact. Our approach set the state of the art on the inaugural COCO keypoints challenge, and significantly exceeds the previous state of the art result on the MPII multi-person benchmark.

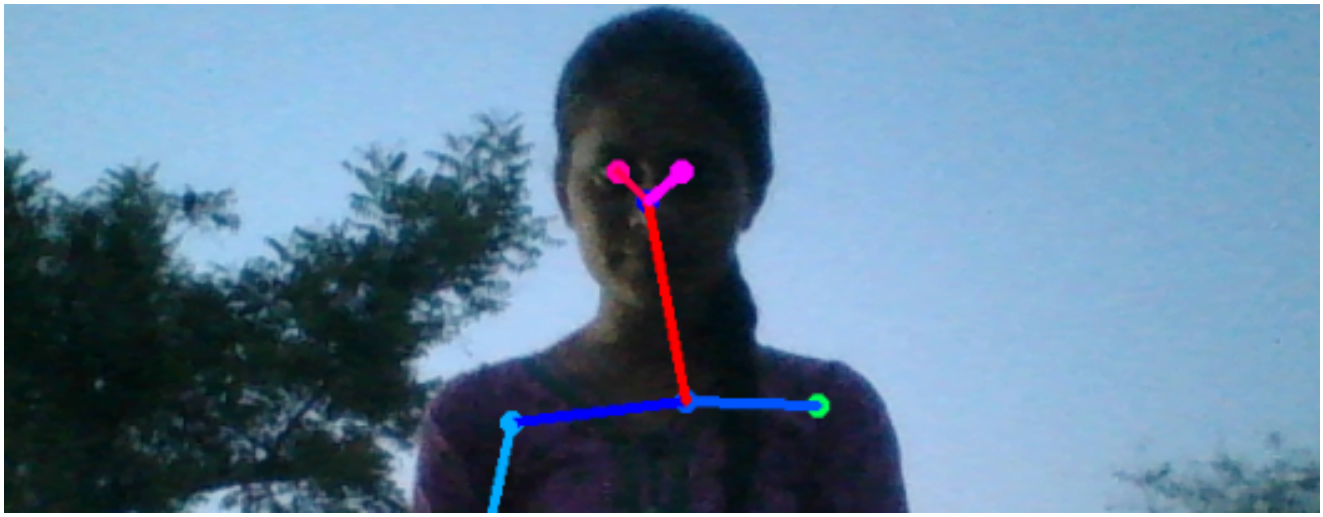
The COCO training set consists of over 100K person instances labeled with over 1 million total keypoints (i.e. bodyparts). The testing set contains “test-challenge”, “test-dev” and “test standard” subsets, which have roughly 20K images each. The COCO evaluation defines the object key point similarity (OKS) and uses the mean average precision (AP) over 10 OKS thresholds as main competition metric. The OKS plays the same role as the IoU in object detection. The reason is that our method has to deal with a much larger scale range spanned by all people in the image in one shot. In contrast, top-down methods can rescale the patch of each detected area to a larger size and thus suffer less degradation at smaller scales.

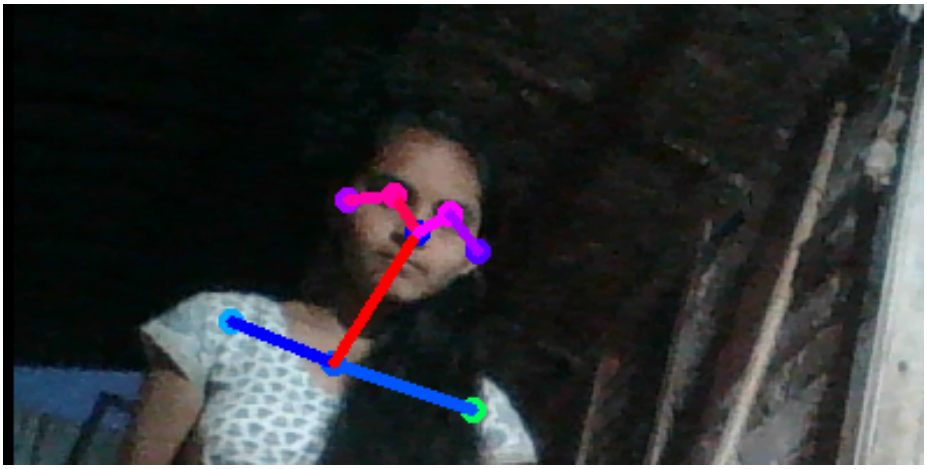
We only update estimations on predictions that both methods agree well enough, resulting in improved precision and recall. We expect a larger scale search can further improve the performance of our bottom-up method. First we need to estimate the keypoints , then we need to join them.

Fig. Estimating keypoints

Fig. Pose estimated by joining the keypoints

5.2 Output screenshots





CHAPTER 6

6.1 Conclusion

Realtime multi-person 2D pose estimation is a critical component in enabling machines to visually understand and interpret humans and their interactions. we present an explicit nonparametric representation of the keypoint association that encodes both position and orientation of human limbs. we design an architecture that jointly learns part detection and association and we demonstrate that a greedy parsing algorithm is sufficient to produce high-quality parses of body poses, and preserves efficiency regardless of the number of people.

We prove that PAF refinement is far more important than combined PAF and body part location refinement, leading to a substantial increase in both runtime performance and accuracy. We combining body and foot estimation into a single model boosts the accuracy of each component individually and reduces the inference time of running them sequentially.

6.2 Future scope

Pose Estimation has huge applications in myriad fields like deep learning, some of the applications are Activity Recognition, for tracking the variations in the pose of a person over a period of time can also be used for activity, gesture and gait recognition. And Motion Capture and Augmented Reality.

pose estimation has many applications in the stream of training robots. Instead of manually programming robots to follow trajectories, robots can be made to follow the trajectories of a human pose skeleton that is performing an action. A human instructor can effectively teach the robot certain actions by just demonstrating the same. The robot can then calculate how to move its articulators to perform the same action.

References

[1] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In 12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: people detection and articulated pose estimation.

[3] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation.