Action recognition using pose estimation

Final Project Part II

Aim of the Project

Aim of this project is to automatically recognize human actions based on analysis of the body landmarks using pose estimation.

Problem Statement

Analysis of people's actions and activities in public and private environments are highly necessary for security. This cannot be done manually as the number of cameras for surveillance produce lengthy hours of video feed every day. Real-time detection and alerting of suspicious activities or actions are also challenging in these scenarios. This issue can be solved by applying deep learning-based algorithms for action recognition.

Project Description

The project has 3 major components:

1. Implementation of CNN based Pose Estimation model

2. Implementation of NN based Action Recognition model

3. Implementation of Action Recognition in videos using pose joints estimated by the CNN model

1. CNN based Pose Estimation model

Unpack and load the pre-processed the dataset with images and joint coordinates. Each pose image contains 7 body joints represented using 14 floating point numbers. Build the CNN model for the dataset and train for few epochs. Test the model with testing set and do inference with single data samples. Save the model. Train a transfer learning model with already trained CNN base model.

2. NN based Action Recognition model

Load the dataset with pose joints as features and actions labels (‘Namaste’ vs ‘Hello’). Split the dataset into training and testing. Train the model for few epochs, test the model, save the model. Analyze the model with new samples of data. Based on the results, flip invariant data augmentation is performed to increase the accuracy where further graph-like features are extracted from the dataset. Training, testing and inferencing is accomplished with the augmented graph features of the dataset.

3. Action Recognition in videos

CNN based pose estimation model and neural networks-based action recognition model is combined to work on images and videos. Action recognition in videos is achieved by processing the frames one by one using the model.

Methodology

Pose Estimation

Input to the human pose estimation model is closely cropped pedestrian or human image. Each training set image has one human inside where pixels are considered as features and target as pair of body joint coordinates. Pre-processed **Pose Estimation FLIC dataset** (http://bensapp.github.io/flic-dataset.html) is be used for this modelling. Set of landmarks from human image can be detected by training convolutional neural networks model with convolution, pooling and fully connected layers with finally landmark point regression as output. Model can be trained based on two strategies:

Strategy 1:

1. Training a model from scratch where model layers can be designed manually, weights of the model will be trained

2. Loading a pretrained base CNN architecture such as VGG16, MobileNet, removing the final softmax layer adding custom layers and training the newly added layers

Strategy 2:

This strategy makes use of “transfer learning” where we can choose whether to retrain the whole network or train only newly added layers. Model can be inferred using a single test sample which is not part of the training/test set, detected pose points can be further plotted.

Action Recognition

Set of human actions can be further recognized using analysis of pose landmarks as features and action label as target. Deep neural networks can be directly trained using dense layers by considering pose point x and y coordinates. A custom dataset with two actions - Namaste, Hello will be used for this task.

Action recognition neural network can be built by directly considering x and y coordinates as feature and action label as target. Since human can appear in any scale in a real scenario, considering raw coordinates as features is not a good idea. This can be solved by considering the skeleton of human pose points as graph, extracting the distance between every joint and further normalizing the distance features. These distance features can be considered for training the action recognition model.

Dataset contains Hello actions performed using right-hand only. Hence, the model cannot recognize a Hello human action performed with left-hand. This issue can be solved by augmenting the action dataset by duplicating the existing actions further flipping the coordinates horizontally. Now with the augmented action dataset, the trained model can detect Hello action performed in both right as well as left-hand. Finally, the pose estimation and action recognition models can be combined by running sequentially. The model can be tested with offline videos and action recognition results will be displayed for each video frame.

Expected Challenges

The project will aim at tackling the following technical challenges:

1. Body landmarks detection of humans from different viewpoints.

2. Detection of human actions which involves pose detection of humans appearing at any scale in the video frame with action performed using either left or right or both body parts.

3. System must meet the real-time processing requirements where the deep learning model must detect pose and actions from video (with 30 fps) at the rate of 33 ms per frame

Dataset Description

Dataset: Frames Labeled in Cinema (FLIC)

The dataset is a collection of 5003 image from popular Hollywood movies. The images were obtained by running a state-of-the-art person detector on every tenth frame of 30 movies. People detected with high confidence (roughly 20K candidates) were then sent to the crowdsourcing marketplace Amazon Mechanical Turk to obtain ground-truth-labeling. Each image was annotated by five Turkers for $0.01 each to label 10 upper body joints. The median-of-five labeling was taken in each image to be robust to outlier annotation. Finally, images were rejected manually by us if the person was occluded or severely non-frontal. We set aside 20% (1016 images) of the data for testing.

How to Start with the Project?

1. Login to the Google Co-lab, load the notebook to the environment. Go to Runtime to choose the “Change runtime type”. For faster training, choose GPU as the hardware accelerator and SAVE it.
2. Import all the necessary Python packages. Numpy and Pandas for numerical processing, data importing, preprocessing etc. Matplotlib for plotting pose joints and showing images, cv2 package for image processing functions, sklearn for splitting datasets, keras for deep learning model creation, training, testing, inference etc.
3. Dataset for Pose Estimation and Action Recognition can be kept inside Google Drive and can be loaded to the Colab. Dataset in the google drive can be accessed using the path “gdrive/My Drive/Dataset_Folder_Name/Dataset_Name.zip”. The pose dataset used in this project is pre-processed FLIC (http://bensapp.github.io/flic-dataset.html) where the Action Recognition dataset is custom made. These datasets can be downloaded and uploaded to the google colab current working directory or it can be kept in google drive which can be mounted to the Colab working directory.
4. Training images and their annotations in action_joints.csv consists of 7 joints - 'left shoulder', 'left elbow', 'left wrist', 'right shoulder', 'right elbow', 'right wrist', 'left eye', 'right eye', 'nose'. The files are read and converted into numpy arrays so that the numpy arrays can be used for training the model. Since the dataset does not provide the validation set (Validation set helps us to monitor the training after each epoch) part of training set can be considered as validation set using function train_test_split. The parameter test_size is the ration between number of training and validation set samples, can be set based on the how bigger we want the validation set to the compared to the training set.
5. From here you can take over to the project and start estimating the human pose

How to submit your project?

1. Share your project via google colab
2. The .ipynb file with details of each step in the markdown
3. Save the model and share the .h5 created

Marks Allocation

1. Implementation of CNN based Pose Estimation model [20 Marks]

2. Implementation of NN based Action Recognition model [20 Marks]

3. Implementation of Action Recognition in videos using pose joints estimated by the CNN [20 Marks]

4. Project report/synopsis with detailed .ipynb [15 Marks]

5. Testing your model on the evaluation dataset [25 Marks]

Note: Evaluation dataset won't be shared with you.