Design Document

Human Activity Recognition

Aaron Finlay, C00226131, 4th year Soft Eng

Supervisor: Greg Doyle

Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY CARLOW

At the Heart of South Leinster

Table of Contents

Introduction

The purpose of this document is to present the technologies that will be implemented during the development of the project. Moreover, the structure of the project and the individual components will be used as an architectural view of the system. The main component from the arsenal of technologies in this project will be the convolutional neural network, however, there is also a haar cascade classifier implementation, should the neural network produce problems and to compare accuracy between the modalities.

Executive Summary

This section of the document will describe the background information to this study. This will entail the problem scenario, the goal of the study as well as a proposal on how to achieve this "goal". Finally, there will also be a motivation piece on why the project was perceived and to what importance it plays in the developer's mind.

Problem

With there being a variety of challenges towards successfully recognizing an activity from a sequence of images, it is useful to outline the variety of problems facing the success of this project. There are a variety of problems which may not be clear from the beginning of this study. Some of the more obvious ones begin with questions such as:

- Where does the data come from, i.e. what test data source is suitable?
- How available is the data you seek, i.e. do you need to create your own dataset?
- Is there enough representation for the different classes useful for classifying an activity in this test dataset?
- Is the data used to evaluate the model actually unseen? i.e. test data rather than training and evaluating the model only on the training data.
- Has the data gone through the correct procedures for data pre-processing, are there any missing values in the data or is the data thoroughly cleansed?
- Is there partial occlusion, fixed viewpoints, cluttered backgrounds within the data? Is the most suitable dataset chosen?
- Are the model parameters or/and the hyperparameters correctly configured in order to give a more accurate model?
- Where is the data/results of previous computations with regards to classifying activities within images being stored?

- Training the model and ensuring that it has been fed enough data is also another question we must solve?
- Is the data “overfitting” or “underfitting” the model, i.e. has the model been trained and tested only by the training set?
- Is this system considered to be adhering to the ethical guidelines in regard to this type of technology with the capability of recognizing humans and their activities and finally, pointing back to the earlier point about where this data is stored, what must also be considered is how is this stored? Who has access to this data, i.e. the confidentiality and just how much integrity of this data is truly kept safe? I.e. could someone later make a change to some historic data, such as when a person was detected falling, could someone tamper with this system to simulate this fall, thus fake falling and fooling the system. Back again to the point “who can access the system”, just how private or secure will this system be?

Goal

The goal Section of this document describes what is hoped to be accomplished such as selecting, integrating, and evaluating a number of different modalities so that an activity within an image is classified and is proven to be as effective as possible. Furthermore, to mitigate against “false-positives, false-negatives” and in essence any type of false output, this is all taken into consideration so that when it comes to split the dataset into its training, test and evaluation counterparts, there is enough representation of data for each segment, i.e. 75% training, 25% evaluation/test and there is an accurate result of what is happening in the image, validating the system works as intended. This will be the epicentre of testing the system, i.e. using unseen data to verify the system can classify accordingly using it’s built up stack of learning from the training set.

This study is mainly a research one, whereby evaluating a number of different models offers the greatest chance of success and by using different ones, should entail the knowledge why each model may differ in accuracy as well as the know how to use the different modalities with different integratable technology. Furthermore, with there being a huge emphasis on the theoretical background of this project, knowing how to mitigate against what is previously mentioned as well as understanding how to perform every step of a typical machine learning project life cycle ensures greater coverage of accuracy on classification as well as a more thorough understanding of every step involved from inputting data, to preprocessing of data, segmentation of data, feature extraction, all the way to recognition.

Proposal:

For achieving the goal of a fully functional recognition system, includes step by step details on how it will be achieved, what techniques/tools will be used, learned and how to evaluate different models

The proposal section of this document entails the “how” when building the system in question. To clearly visualize how each aspect of this system will be dealt with as an individual task, composing of a greater collective of objectives. This will include:

- Selecting the datasource - upon analysing the research to date, the kinetics dataset offers the most viable dataset with there being plenty of data publicly available covering many class types such as human eating, human running, human falling - which is the main use case scenario described throughout this study. This dataset has no missing values, the data is cleansed, it is ready to be integrated with a training script to train the model.

- Choosing the "right" model - from the data gathered thus far, the use of convolutional neural networks seems to be the best approach when dealing with the learning of the model, which is paramount in the system being able to detect "unseen" data.
- Learning the theory - from intense research into the matter, the learning process of how neural networks "learn" has been understood, and how to correctly configure model parameters and hyperparameters have also been investigated which will equip a developer with the tools to ensure higher accuracy of the model.
- Splitting the dataset - when accessing a dataset, typically it is best to divide your data into specific sets - train data used to feed data to the model, test data which is used to evaluate the model. If the same data spent on training the model is used to evaluate the model, then you would never even realize you have "overfitted" your model. The model should be judged on its ability to "predict" new, unseen data, e.g. 75% training data, 25% test data.
- Model training - the bulk of the duration of this project will be spent on this step. This time is typically spent on exploring the data, cleansing the data and engineering new features.
- Comparing test vs training performance - this will prevent overfitting, i.e. if the model performs very well on training data, but poorly on test data, it is then considered to be overfit. Tuning models refers to tuning hyperparameters, e.g. how many neurons should be in this layer inside the entire neural network. There are two types of parameters in machine learning algorithms, that is, model parameters and hyperparameters. Key distinction -> model parameters can be learned directly from the training data meanwhile hyperparameters cannot.
    - Model parameter attributes:
        - Learning attributes that define individual models.
            - Example, regression coefficient.
            - Another example, decision tree split locations, these can all be learned directly from the training data.
    - Hyperparameter attributes:
        - Express "higher-level" structural settings for algorithms.
            - Example, strength of penalty regression used in regularized regression.
            - Another example, the number of neurons to include in a layer within a neural network.
            - These decisions are made prior to fitting the model because they cannot be learned directly from the data.
- Cross-Validation - this concept will help tune the models chosen. It is a method used for getting a reliable estimate of model performance using "only" training data. There are several methods for enacting cross-validation, the most common of which, is 10 fold cross-validation. This way breaks the training data into ten equal parts also known as folds, essentially creating 10 miniature train/test splices.
    - Steps for 10 fold Cross-Validation:
        - 1) Split data into 10 equal parts or folds.

        - 2) Train the model on 9 of the 10 folds.
        - 3) Evaluate/test your model on the 1 remaining fold - also known as the "holdout fold".
        - 4) Perform steps 2 and 3 10 times, each time holding out a different fold.
        - 5) Average the performance across all 10 hold-out folds.
    - The average performance across all 10 hold-out folds is the final performance estimate also known as the cross-validated score. Because of the 10 mini train/test splits, this score is usually pretty reliable.
- Fitting and tuning the model - In this step, all that is required is to perform the entire cross-validation loop detailed above on each set of hyperparameter values that are desired to be tested.
    - **Pseudocode for cross validation:**

```
For each algorithm (i.e. Convolutional Neural Network - CNN, etc.)

    For each set of hyperparameter values to try:

        Perform cross-validation using the training set
            Calculate Cross-Validated score
```

- Upon completion of this process, you will have a detailed cross-validated score for each set of hyperparameter values, for each algorithm. Then you pick the best set of hyperparameters within each algorithm.
    - For each algorithm:
        - Keep the set of hyperparameter values with the best cross-validated score. You then re-train the algorithm on the entire training set (without cross-validation).
- Loss function - Cross-entropy loss, also known as log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss essentially increases as the predicted probability diverges from the expected label. Therefore, a prediction that the probability is of .010 meanwhile the actual label observing 1 would therefore be bad and ultimately, the result would end in a high loss value. What would be considered the perfect model is one whose value is at a log loss of 0.
    - Implementation Code:

```
def CrossEntropy(yHat, y);
    If y == 1:
        Return -log(yHat)
    Else:
        Return -log(1 - yHat)
```

    - Notated in Maths would be as follows:
      In binary classification, where the number of classes NM equals to that of two, then cross-entropy loss can calculate as follows:

-(ylog(p)+(1-y)log(1-p)) //formulae

If M>2 (which means a calculation based on multiclass) a calculation based on loss for individual class labels can be performed and retrieval of observations and result equation summarizing the result.

- Select Winning Model - Because the test set of the data is unseen, it can now be used to get a reliable estimate of each model's varied performance.
- Performance metric - In this instance, for tracking the performance of a classification task such as the one at hand, Area Under ROC Curve (AUROC) would seem suitable. When attempting to check or visualize the performance of the multi-class classification problem, AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve. ROC is a probability curve and AUC represents the degree or measure of separability.

Motivation

With there being a chance that if sufficient work is made, it could be used for people in the real world, it would accomplish one of the primary goals of the project, which is making an impact. Furthermore, Ireland has an ageing population and any contribution that can be made in assisting elderly folks then it is also another motivation factor. Finally, the number one motivator for this was the idea of enabling a more independent lifestyle for elderly people, removing the need of wearable sensors.

Moreover, another benefit was the length of learning the process, benefitting me in this area. With no prior experience or knowledge in machine learning and in activity recognition systems, the learning curve is steep but also a rich journey which will add an arsenal of skills in data science and machine learning.

GUI Explanation

As this project is heavily research focused and based around understanding and applying machine learning tools, techniques and technologies, the scope of the project as per planned is to build a proof of concept system. This system proposed will contain the functionality to be able to isolate, classify and thus recognize an image and an activity that occurred through a sequence of images via a video stream. This system therefore has little user interaction, and thus little GUI features to be described in detail.

If the system goes ahead of schedule and sufficient work is achieved it could be possible near the end to consider wrapping it up as an app. However, due to the depth of research involved and since there is practically little to no knowledge in the area of machine learning upon embarking on this project, this alone makes this project a challenge. The coding of this system won't be the most difficult part but the tuning of it and the configuring of hyperparameters, segmentation of data, and the in-depth preprocessing of data that is always cumbersome. As per mentioned earlier, the basis of this docund the algorithm itself.

Data Source(s)

The data source chosen to train the model is the Kinetics 700 action class dataset. This contains thousands of examples of varied human activity, such as falling that is the main focus of this study. There are varied viewpoints, reduced background clutter, good quality and good representation for falling along with other activities. However, although this dataset gives a good starting point to training and evaluating the model, the goal is to produce data that can be classified, without being visible (also known as unseen/test data) to the model.

Object Oriented Design

This section of the document will describe the design of the system in an object oriented fashion. A full detail of the class and sequence diagram(s) will be provided, showing the coupling & cohesion of the classes, as well as the features involved with the system within the sequence diagram.
Update: It was found that since there is little user interaction with the system, there is more emphasis on the algorithm/model. Therefore, a section dedicated to the model will be described in detail.

Plagiarism Declaration

INSTITUTE *of* TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) AARON FINLAY

Student Number(s): C00226131

Signature(s): Aaron Finlay

Date: 30th of April, 2021