Classification of social gestures

Recognizing waving using supervised machine learning

SVANTE ROLLENHAGEN
Klassifikation av sociala gester

Igenkänning av vinkning med hjälp av övervakad maskininlärning

SVANTE ROLLENHAGEN
Abstract

This paper presents an approach to gesture recognition including the use of a tool in order to extract certain key-points of the human body in each frame, and then processing this data and extracting features from this. The gestures recognized were two-handed waving and clapping. The features used were the maximum co-variance from a sine-fit to time-series of arm angles, as well as the max and min of this fitted sinus function. A support vector machine was used for the learning. The result was a promising accuracy of 93% ± 4% using 5-fold cross-validation. The limitations of the methods used are then discussed, which includes lack of support for more than one gesture in the data as well as some lack of generality in means of the features used. Finally some suggestions are made as to what improvements and further explorations could be made.
Sammanfattning

Contents

1 Introduction 1

2 Background/theory 1
   2.1 Data collection .................................................. 1
   2.2 Pre-processing ..................................................... 2
   2.3 Feature extraction ................................................. 2
   2.4 Training and testing .............................................. 2

3 Methods 2
   3.1 Data collection .................................................. 3
   3.2 Pre-processing ..................................................... 3
   3.3 Feature extraction ................................................. 5
   3.4 Training and testing .............................................. 15

4 Results and discussion 15

5 Conclusion 17

6 Acknowledgments 18
1 Introduction

There have been several advancements in the last decade in recognizing gestures from video. In this paper existing technology within recognizing of human limbs and posture is used to predict complex gestures, in this case waving. The goal of this study is so to recognize waving from video data, more specifically determine if a video is containing either waving, or clapping. (Waving in this context means a two-hand waving over the head, see figure 1) [12].

One possible application for these kinds of technologies is robotics. With "smart" applications today being able to, more or less humanlike, talk with a human through text or even speech, it is also wanted for robots to be able to recognize human posture and gestures, both for long distance communication as well as communication through sign language.

Another, perhaps more direct, use of these kinds of developments is for surveillance cameras to be able to detect people in need. Related research has for example been conducted for detecting falling people from video data [8], recognizing hand sign language in real-time [11] as well as several other applications. A related work is one performed by a team at KTH where they used local space-time features in order to recognize gestures, including hand waving and hand clapping [2]. Summaries of more work done in this area can be found, for example, in [7] and [12].

2 Background/theory

Machine learning is a very broad field, with many techniques and different approaches. However, it can roughly be broken down into the following steps: data collection, pre-processing, feature extraction, model training, and finally testing. The basic theory behind these is presented below:

2.1 Data collection

The first step in any machine learning is the collection of data. This can be in any form; raw text, images, videos, tables of numerical values, etc. How much data is needed depends greatly on the situation and what kind of data it is. More data is (almost) always better, it is however often difficult to find or sample large sets of data. Especially when it comes to video data is is quite difficult to find large datasets compared to text samples for example.
2.2 Pre-processing

Pre-processing consists of reducing the data to lower dimensional and/or more relevant data. This is done for a couple of reasons: first of all it reduces the size of the data so that the later steps can be performed faster (or at all). Secondly, if done correctly, it narrows down the data into its more relevant parts. Within the field of video data, one can processes it by finding the contours of the person(s) in it, and then reducing the background to a single color, as well as the body of the person. Another approach that can be made is finding point trajectories of the subject(s) in the video. This will remove much unnecessary data since the colors and shapes of the background or cloths (usually) are not relevant in the applications [12].

2.3 Feature extraction

Extraction of features is the final step before training a model; and a very important one. A feature in this context is simply a scalar or vector that is calculated for each data sample, and that in the best of ways encapsulates the most relevant information in the data. When dealing with gesture recognition this could be the velocities of detected bodyparts in the pre-processing, joint angles, or taken from more abstract entities such as optical flow (the motion of the pixels in a video) [6] [12].

2.4 Training and testing

Finally, training means using some Machine Learning algorithm with the features extracted from the dataset to train it, and finally test it on some part of the data that was not used for the training. This is done so that this data is not already accounted for directly in the model, which is known as "overfitting". A common algorithm to use for gesture recognition (among other applications) is a Support Vector Machine (SVM) (this also happens to be the algorithm used in the project that created the dataset used). [2] More detailed information on SVMs can be found in [3].

3 Methods

Described in the following is the number of steps taken to get from the raw video data to features put into the machine learning algorithm used.
3.1 Data collection

The data used was from a dataset of six types of human actions, performed by 25 subjects a number of times [2]. Since a large amount of data is needed, it was taken from a dataset from an earlier research project instead of collected manually. The videos used are the ones containing waving and clapping respectively. The videos are 160 times 120 pixels, greyscale and are around 20 seconds each. The videos have different backgrounds, the subjects are wearing a range of different clothing and in a number of the videos the camera is zooming in and out throughout the clip. Figure 1, as well as Figure 2 displays 4 sample frames from one such clip of waving and clapping, respectively.

3.2 Pre-processing

In this case, an opensource project [1] [14] was used for recognizing limbs and certain key-points of the human skeleton. This was done in order to reduce the
Figure 2: 4 sample frames of clapping video.
Figure 3: Key-points, the green ones is the ones extracted.

data from all pixels, in all frames, to a total of 18 points per frame. That means
for each frame we have X and Y coordinates for the key-points used, so a total of
36 values for each frame, compared to 20480 when using 160x128 black and white
footage. (For higher resolution, this is obviously much higher). These points and
their locations in four different frames are shown in Figures 4 and 5.

In the case of waving, it is clear that the positions and movements of the legs,
head or torso are not really relevant as to if someone is waving. (You can wave
when sitting, standing, walking etc.) Therefore only the key-points corresponding
to the center, both shoulders, elbows and hands were extracted. Figure 3 displays
the key-points extracted by OpenPose as colored circles, and the ones used here
as green circles.

3.3 Feature extraction

After OpenPose extracted the key-points for all videos, Python and NumPy were used for the rest of the data processing and feature extraction.
Figure 4: Extracted key-points from OpenPose in four different frames of one waving video
Figure 5: Extracted key-points from OpenPose in four different frames of one waving video
At this point, it is clear that the absolute positions of the relevant key-points (the green circles in Figure 3) is not as relevant for the posture of the subject as the angles in between them. This is the case for a couple of reasons: first, it should not matter if the camera is tilted, or the person tilts to the side while waving. Secondly, changing the zoom level or having the subject at different distances from the camera should not change whether or not (s)he is waving.

Therefore, the relative angles in the elbow, as well as armpit, were calculated for each frame. These are shown in Figure 6. In order to calculate these angles, a bit of care is needed. A naive attempt could be using the definition of the scalar product for vectors, namely:

\[ a \cdot b = ab \cdot \cos(\theta) \]

where \( \theta \) is the angle between vectors \( a \) and \( b \). Solving for \( \theta \) leads to the formula

\[ \theta = \arccos \left( \frac{a \cdot b}{ab} \right) \]

where \( a \) and \( b \) would be the difference vectors for the position vectors of the points. However, this formula can lead to high errors when the vectors are close to parallel.
[13], so a better formula is instead:

\[ \theta = \text{atan2}(\|a \times b\|, a \cdot b) \]

where \( \text{atan2}(x, y) \) is the multi-valued inverse tangent function, which by taking the cross product and the dot product of two angles as input returns an angle which, based on the signs of the inputs, correctly determines the quadrant of the angle.

At this point, the time-series for each of the clips was stripped down to the first 200 frames. This was partly done in order to make all the time-series of equal size, and partly because it is more realistic that a subject in real life is waving for a couple of seconds than for 20 seconds straight.
Figure 7: Time series of the extracted angles, for one sample video.
Figure 8: Time series of the extracted angles, for one sample video.
If the raw angle-time-series were to be used as features, the results would be poor, since this assumes that all waving was started at the same time, and also done equally fast. A video sequence compared to the same sequence but slowed down should lead to the same (or at least similar) features. The same should be if one were to cut away a small portion of the beginning or end of the clip. To solve this, the data needs to be narrowed down even further. If one would compare the graphs in Figures 7 and 8, it seems like the pattern in Figure 7 looks very sinusoidal, while Figure 8 not as much (while still equally periodic).

Another interesting fact is that the angles of the shoulders in Figure 8 never seem to go above $-40^\circ$, while the ones in Figure 7 varies from $-60^\circ$ to $+80^\circ$. This is very natural, since the subjects clapping in the data did so mostly below the head, while the waving was performed above the head. (see Figures 1 and 2) Similar patterns could be seen in the other samples in the data. Because of this, a fitting function was used to fit the angle-time-series to a sine function. Figures 9 and 10 shows one such fitting for a waving video and a clapping video, respectively.
Figure 9: Time series of the extracted angles, for one sample video.
Figure 10: Time series of the extracted angles, for one sample video.
<table>
<thead>
<tr>
<th>run</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.95</td>
<td>0.95</td>
<td>0.925</td>
<td>0.897</td>
<td>0.947</td>
</tr>
</tbody>
</table>

Table 1: Accuracy from 5-fold cross-validation for waving recognition using an SVM

3.4 Training and testing

Three values were extracted from each fit (as explained in 3.3), and later to be used as features. The first one is the max-covariance, a measure of how "good" the fitting is [10]. This was done since the waving time-series seemed more sinusoidal than the clapping ones, as can be seen in Figures 7, 8, and 10. The other two values used as features was the minimum as well as the maximum of the fitted sine function. This was done since it is clear that the extreme values of the angles (especially the ones of the armpits) was rather different for waving and clapping.

Finally, these features were inputted to a built-in function in the SciKit learn library for python, namely a Support Vector Machine (SVM) [9]. Since the dataset is relatively small, cross-validation was used to work around this fact. This means that the data was split into 5 different sets, and then the model was trained on 4 of them, and then evaluated on the fifth one. Doing this 5 times, using different ones each time, results in 5 different accuracy measurements, and the mean of them is regarded as the accuracy of the model [5].

4 Results and discussion

Using the features described in 3.3, a Support Vector Machine (SVM) was trained using 5-fold cross-validation, resulting in the accuracy presented in Table 1. This results in the final accuracy of 93% ± 4%, where 93% is the mean of the values from Table 1 and ±4% is the standard deviation.

At first glance, the results look very promising; over 90% accuracy with just 200 samples. However, there is a number of things to consider: first of all, the approach in this report is rather specialized to waving. If one were to perform the exact same procedure on recognizing people running instead on waving for example, the results would probably be very poor since only the angles at the arms are considered. However, with some generalization a very similar approach could be done to most gestures:

First, use OpenPose, or a similar tool or method, in order to extract key-points of the body in a video. Then, calculate not only the angles in Figure 6 but also leg
angles, head angle etc. After this, plot the times-series of the different angles, and try to see what function could be fitted to it. In the case of waving, sine-functions seemed like the best fit; for other gestures, other functions could be used. Then fit the angle-time-series to this function, and use the maximum covariance as well as some other parameters of the fitted function (for running the frequency could be important to distinguish it from jogging, for example). Some work needs to be done to select appropriate functions as well as the parameters to use as features, but it could very well be done, and the results from this report are promising for this.

Another limitation of using this method is that it is assuming that the gesture performed (whether waving or clapping) is the only thing occurring in the clip. Since the method includes fitting a sinus function to the time-series of the arm angles, and the error of this fit is used as a feature, if anything else were to happen before any waving, this would make the fit poor, and thus make poor decisions based on it. A naive approach to try to work around this is to split up each clip in 5-second clips, and then perform the same feature extraction on each clip. This could take a lot of time, but would at least fix the issue of other things in the video interfering with the feature extraction.

One potential problem is that this method only uses 2D data. When a limb is pointed towards the camera, it is difficult to recognize the limbs, and the angles in between them effectively collapses. This is also due to the angles being the thing used. This can be solved by using 3D data, such as accelerometers, depth cameras or more than one camera used at the same time. From this, a 3D skeletal model of the body would be extracted, and then angles could be extracted just the same way as in the 2-dimensional case. However, there are two major arguments for using 2D data over 3D, the first one being simply that using 2D seems to work fine, both in this study as well as in related works. Another reason is that most of the "real" data in the world is 2D rather than 3D. Both surveillance cameras and peoples smart phones are producing 2D data, and it makes more sense to use what is available than to complicate things when it does not seem to be needed.

An important point to make is that the whole pipeline fully depends on OpenPose (or a similar tool) successfully finding the key-points necessary. If it were not to, the whole thing naturally falls apart. Luckily, in the data used here, OpenPose was almost always successful in finding the key-points needed, even with the low resolution of the video. In some of the videos where the clothes were of a very similar color as the background, it misinterpreted the flap of a jacket as the arm for a few frames, but other than that it did its job very well. This means that
even with bad cameras, this approach will work, as long as the tool used for limb-recognition is functioning.

One possible issue with the data used is that all of the videos have the subject faced more or less directly at the camera. Even though the methods used takes into account rotation of the camera, different distances to the subject as well as zooming in and out, it does not consider if the subject is turned with its side to the camera, even partly.

Finally, OpenPose is very heavy processing-wise (running it on a standard laptop required resolution set to as low as 320x240 in order to run in real-time with a webcam). On the other hand, since the training will be performed in advance, it does not matter as much that it takes a lot of time, and in the applications with security cameras would use with similarly low resolution. Also, the approach made in this report does not depend on the exact tool used to extract the key-points, only that they are somehow extracted from the video(s). As more refined methods on finding such key-points is developed, the whole process would be sped up. As mentioned in the Introduction, a lot of work is nowadays done with more direct methods on the video data, rather than first extracting key-points. The results here however, combined with the intuitive concept points towards key-points extraction still having a role to play in gesture recognition.

5 Conclusion

The goal of this project was to recognize waving from video data, and to distinguish it from clapping. This was done by using a tool (OpenPose) to extract certain key-points of the human body in all frames of videos of people waving and clapping respectively. After this, angles of the arms were calculated from these key-points, and it was by eye identified that the ones with waving were more sinusoidal than the ones with clapping. A sine-function was hence fitted to the angle-time-series and an estimate of how good of a fit it was, together with the maximum and minimum of it, were used as features inputted into a Support Vector Machine. Using 5-fold cross-validation resulted in an average accuracy of $93\% \pm 4\%$. This is a high value, but do come with some limitations. One such is that the data used were much more simplistic and clean than one could expect "real" data would be; it only contained one subject in each video, looking more or less straight into the camera and only performing one gesture throughout the whole video. It is also described a possible generalization of the method used in order to be applied to more gestures than the ones analyzed here.
6 Acknowledgments

First, I would like to thank my supervisor Iolanda Leite, at the department of robotics, perception and learning, for inspiration, help and support throughout this whole project. I would also like to thank Hannah Haapanen for designing the poster used at the presentation on the 14th May 2018.
References


