SMILE RECOGNITION

Jesper Granat, Peter Todorov

jesper.granat@tuni.fi, peter.todorov@tuni.fi

ABSTRACT

In this study, a convolutional neural network was trained to recognize smile from an image. The training and deployment was done using the Keras framework on top of the Tensorflow framework. The training data was 4 000 images of smiling and non-smiling faces. The accuracy on this dataset was 86,XXX %.

1. INTRODUCTION

Since their introduction [REFERENCE], Convolutional Neural Networks (CNNs) have been widely used in solving various image processing problems, such as classification, object recognition and image enhancement [REFERENCE]. In this study, a custom design CNN is trained for real-time smile recognition. Real-time in this context refers to inference speed where the network can be run on live video feed in over 20 fps on a laptop with GPU.

2. BACKGROUND

Convolutional Neural Networks (CNNs) have been shown to be essential in modern computer vision applications, such as classification [REFERENCE]. Classification in particular is rather difficult problem due to its complexity. Here, the goal is to assign a label from a predefined set of labels to an image. Difficulties in this task include truncation of objects (object is partially not in the image), occlusion of objects (some other object is blocking some parts of the desired object), variations in color and lighting, and intra-class variation. The complexity of classification can be intuitively explained by considering a case where an image of a cat is desired to be classified. The problem here is learning a representation of a cat, as many cats can look different, an image of the same cat from different angle and lighting yields in a completely different matrix of pixel values in the image. [REFERENCE]

Traditional approaches to image classification have utilized hand-crafted algorithms such Histogram of Oriented Gradients (HOG) [REFERENCE]. In such algorithms, pixel color values and pixel proximities are utilized to create a representation of the training data. Since 2012 [REFERENCE], CNNs have been dominant in classification. Particularly, deep CNNs from 2015 [REFERENCE] are still dominant in classification. These can include hundreds or even thousands of layers of convolutions. Typical CNNs for classification include two types of layers: convolutional layers and pooling layers. Pooling layers introduce non-linearity to the model to make is feasible to include several layers of convolutions. Convolutional layers, on the other hand, are responsible for creating the image representation. The last layer(s) in CNNs are usually fully connected layers. The output of the network are the probabilities of each class being in the image.

CNNs for classification are trained with cross-entropy loss [REFERENCE]. The idea of the loss is to punish weights that are responsible for wrong classification, and reward weights that are responsible for correct classification during training. Additionally, batch normalization [REFERENCE] is used to generalaize the results. Furthermore, classification with CNNs includes many more bells and whistles that create the absolute best results.

3. SOLUTION

4. RESULTS

5. CONCLUSION