Image Classification for Personal Protection Equipment (PPE)

Mohammad Najib Bin Abdul Rahim, G1923437

I. Introduction

Image classification is one of the most commonly used application of Machine Learning (ML) in various industries. There are various types of classifiers can be used which receive images as input and label images with corresponding class as output. Deep Learning (DL) is the most commonly algorithm these days, specifically under Convolutional Neural Network (CNN).

For the main purpose of developing exposure to various image classification algorithms, this project focuses mainly on utilizing the Bag of Words (BoW), Fully Connected Neural Networks (FCNN), Convolutional Neural Networks (CNN), Transfer Learning (TL), and fine-tuning method. The same datasets are being used throughout this project.

For each method, several variations of the algorithms are applied in order to analyze the corresponding changes to the trained model. This is to enhance understanding on the performance of the algorithm and its behavior with changing of parameters.

II. Dataset

The datasets utilized for this project comprises of mandatory Personal Protective Equipment (PPE) being used mainly in any Oil and Gas (O&G) plant, offshore or onshore. It covers safety boot, safety glasses, glove, coverall, and safety helmet. The images are initially scraped from google images for 100 images per classification. By using image augmentation techniques, specifically applying rotation, translation, transformation, and flipping, the number of images are increased to 1440 images per classification for training data and 50 for test data.

III. Classification Algorithms

3.1 Bag of Words (BoW)

Bag of Words or also known as Bag of Visual Words (BOVW) when applied to images is one type of image classifier. Image features are considered as the words which represent image as a set of features consisting keypoints and descriptors. Keypoints are the “stand out” points in an image and descriptor is the description of the keypoint. Those are used to construct vocabularies and represent each image as a frequency histogram of features that are in the image. The frequency histogram later used to predict the category of unidentified image.

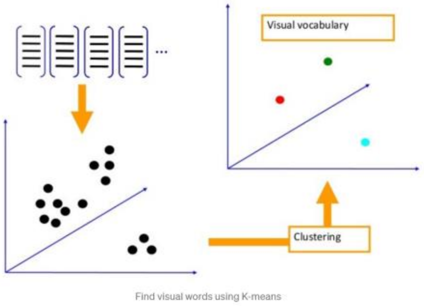
Some of the feature extractor algorithms that can be used are SIFT, local HOG, and SURF. The descriptors are then clustered using clustering algorithm such as K-Means and GMM. The center of each cluster will be used as the visual dictionary’s vocabularies.

Frequency histogram is created for each image from the vocabularies and the frequency of the vocabularies in the image. Those histograms are Bag of Visual Words (BOVW)

K-Mean clustering algorithm starts by choosing K points as the number of clusters to represent the dictionary’s vocabularies. Each K is the centroid of each cluster. Then all descriptors will be assigned to the closes centroids. Next step will iteratively recompute the centroid of each cluster and stop when centroid does not stop.

To encode novel images, we find the frequency (histogram) of local descriptors associated to each visual vocabulary (centroid of k-means). Each image then is represented by a k-dimensional vector. Each value in the vector is the number of descriptors assigned to the cluster.

The classifier SVM which is one of the most robust prediction methods that analyze data for classification and regression analysis. It receives the training images and apply a linear classification to the different labeled features. The idea is to maximize the width of the gap between the categories. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.



3.2 Fully Connected Neural Networks (FNNs)

FNN is the simplest form of artificial neural network. FNN uses the deep learning to perform classification.

In this network, the information moves in only one direction—forward—from the input nodes (neurons), through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. We can see the structure of FNN in figure (2). IN our case (image classification), the input is an image which is flattened to be fed through input layer and the output represents the required classes. FNNs primarily used for supervised learning. That is, feedforward neural networks compute a function f on fixed size input x such that f(x)≈y for training pairs (x,y).

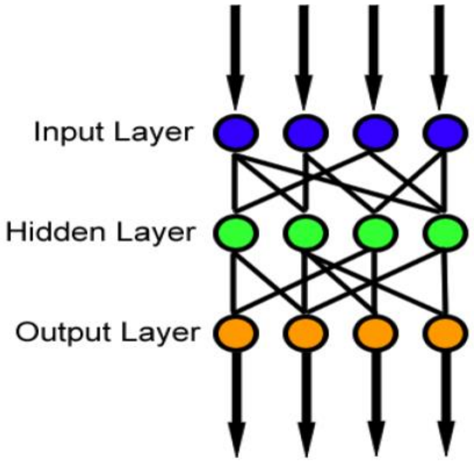


Figure 2: Fully Connected Neural Network (FNN)

The basic unit in NN is the neuron. An artificial neuron takes on an input vector and outputs a scalar value. The neuron is parameterized by a set of weights. Each weight is used as a multiplier for a scalar input. The output of the neuron is the result of applying a nonlinear activation function on the sum of the weighted inputs. Thus, a neuron with weights w, inputs x, output y, and non-linear activation function f is represented as:

$$y = \emptyset\left((\sum_{i=0}^{n} w_i \cdot x_{i)+bias} \right)$$

Where n is the number of layer inputs, x is the input and w is the associated weight. Bias is the neuron's controlling value which decide if this neuron is going to give an output or not. Our purpose is to learn the weights and the biases to have a correct output or mathematically to reduce the output error. Since the small change in weights and biases may lead to a big change to the output, non-linear activation functions f (x) are applied to the neuron term. An additional requirement of the learning algorithms over f (x) is differentiability. Some of the nonlinear activation functions are sigmoid, ReLU and Tanh.

An artificial neural network (ANN) consists of a set of connected neurons. Typically, neurons are grouped in layers. Each layer takes a set of inputs for computing a set of outputs. The input-output relation is determined by the weights in the layer.

DNNs are usually trained using supervised learning. The objective of the training procedure is to find the network parameters that minimize a loss function. The approach normally used is gradient based approach which iteratively optimize the output. The output of the network is back propagated through the network layers to update the neuron's parameters to have the minima loss based on gradient based method.

For image classification task, input is an image and output is the label or class of the image. The process of the training is the same as mentioned before where the ultimate training ensures the class of the image is correct. We can play around with some parameters in the algorithm to enhance the training process such as changing the learning rate, choosing different non-linear activation function or maybe using dropout technique.

Dropout technique consists of using only a random subset of the neurons in each layer (along with its connections) during training episodes to reduce overfitting. This procedure able to improve the generalization error of the networks.

3.3 Convolutional Neural Networks (CNNs)

CNNs which are under Deep Neural Networks category able to extract various features from the image and differentiate one from the other.

CNNs typically consists of an input and an output layer with multiple hidden layers depending on the architecture. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a ReLU layer, and is subsequently followed by additional convolutions such as pooling layers. Fully connected layers are connected sequentially with the last convolution layer. CNN referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.

The CNN classification algorithm is similar to FNN in taking an image as an input, feedforward, computing the loss and then backpropagating the loss to update the parameters and finally reaching the minima loss. The difference is in the architecture of the network. Instead of having the input image as number of pixels, ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

(a) Convolutional Layer

The input of the CNN is an image with shape (image height) x (image width) x (input channels). Convolutional layers convolve the input and pass its result to the next layer. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter,K. We select K as a nxnx1 matrix. The filter is applied to the image and progressively convolve the input image and move from up left to down right in steps of the stride length until it finish the process. The output is a feature map with the size of nXnXnumber of filters. Where

n = [(W−K+2P)/S]+1 .
W is the input width, (32)
K is the Kernel size (3)
P is the padding (0)
S is the stride (default = 1)

The convolution operation brings a solution to this problem as it reduces the number of free parameters (weights), allowing the network to be deeper with fewer parameters. For instance, regardless of image size, kernel of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters.

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network, which has the wholesome understanding of images in the dataset, similar to how we would.

(b) Pooling Layer

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling - figure (5) returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

(c) Classification – Fully Connected Layer (FC Layer)

The fully connected layer receives the output from last convolution layer as an input. The input image is flattened into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

3.4 Transfer Learning

In CNN, we can apply the knowledge learned in previous task to novel ones. In Transfer Learning we take advantage of the very well-trained models to perform a task (classification in our case). The benefit of this is to get faster performance and accurate result. In addition, less training data is needed to do the training. So instead of learning a CNN from scratch we just use the pre-trained model and feedforward our dataset. The only adjusting is that the last layers (the ones that make the final classification) are replaced and whose first layers are re-used. The same features can be used for solving the two different problems. And that the classifiers (output layer) need to be trained again only while freezing the learning of parameters of the reused layers.

3.5 Fine-tuning

In fine-tuning, the same concept of Transfer learning is applied except we need to train the pre-trained parameters more as well as the new used layer(s). So the new layers must always be trained by using the second set of objects. Also, the reused layers can be trained further with a smaller learning rate depending on data availability. In general, we can set learning rates to be different for each layer to find a trade-off between freezing and fine-tuning.

IV. Results

4.1 Bag of Words

4.2 Fully Connected Neural Networks

4.3 Convolutional Neural Networks

4.4 Transfer Learning

4.5 Fine-tuning

V. Conclusion