Fifth Dimension Technical Report

Fifth Dimension Deep Learning Algorithms for Unstructured Data: Overview and Benchmarking

Document #5D-16-000

THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO FIFTH DIMENSION HOLDINGS LTD. ("FIFTH DIMENSION") AND IS BEING SUBMITTED TO YOU AS A PROSPECTIVE CUSTOMER OF FIFTH DIMENSION SOLELY FOR YOUR CONFIDENTIAL USE WITH THE EXPLICIT UNDERSTANDING THAT, WITHOUT THE PRIOR WRITTEN PERMISSION OF FIFTH DIMENSION, YOU WILL NOT RELEASE THIS DOCUMENT OR ANY PORTION OF THIS DOCUMENT, DISCUSS ANY INFORMATION CONTAINED HEREIN, OR MAKE REPRODUCTIONS OF OR USE THIS DOCUMENT OR THE INFORMATIN CONTAINED IN IT, FOR ANY PURPOSE OTHER THAN THE EVALUATION OF A POTENTIAL PURCHASE OF FIFTH DIMENSION'S PRODUCTS AND SOLUTIONS.

THE DISCLOSURE AND PROVISION OF PROPRIETARY INFORMATION UNDER THIS DOCUMENT BY FIFTH DIMENSION SHALL NOT BE CONSTRUED AS GRANTING TO YOU ANY RIGHTS WHETHER EXPRESSED OR IMPLIED BY LICENSE OR OTHERWISE ON THE MATTERS, INVENTIONS OR DISCOVERIES TO WHICH SUCH PROPRIETARY INFORMATION PERTAINS OR ANY COPYRIGHT, TRADEMARK OR TRADE SECRET RIGHTS. ALL SUCH PROPRIEATARY INFORMATION SHALL REMAIN AT ALL TIMES THE EXECLUSVIE PROPERTY OF FIFTH DIMENSION.

THIS DOCUMENT DOES NOT PURPORT TO BE ALL-INCLUSIVE OR TO CONTAIN ALL THE INFORMATION THAT A PROSPECTIVE CUSTOMER MAY DESIRE IN ITS EVALUATION OF FIFTH DIMENSION'S PRODUCTS AND SOLUTIONS.

WHILE FIFTH DIMENSION BELIEVES THAT THE INFORMATION CONTAINED HEREIN IS ACCURATE, IT EXPRESSLY DISCLAIMS ANY AND ALL LIABILITY FOR REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED CONTAINED IN, OR FOR OMISSIONS FROM, THIS DOCUMENT OR ANY OTHER WRITTEN OR ORAL COMMUNICATION TRANSMITTED OR MADE AVAILABLE.

Table of Contents

Table of Figures

1 Introduction

We report our current results in three major application domains, addressed by the Fifth Dimension Research & Development group: computer vision (object and face recognition), speaker recognition and document similarity. Results are reported with respect to:

a) Well-known and publically available benchmarks, and
b) Our own scoring methods, derived from our forecasted use-case in an operational environment.

The standard benchmarks, as well as our ad-hoc scores, are described in detail in this document. In each problem domain, our solution is outlined, providing the main stages and framework, followed by the details of our results. The scope of this document, however, does not include the details of the method such as network architectures, specific similarity functions chosen or implementation technicalities. The rest of this report is organized as follows: Section 2 reviews some general perquisites and frequently used concepts. Section 3 reports our results on selected computer vision problems, Section 4 reports our results in speaker recognition problems, Section 5 reports our results in text similarity problems, and Section 6 concludes this report.

2 An Overview of the Deep Learning Paradigm

This section provides a brief review of some basic concepts and theoretical background. While some use of undergraduate level mathematical notations shall be made, we aim to provide a document that is readable and understood to a non-professional with little technical background. We skim through perquisite material that can be found in standard text on machine learning and statistical methods (for example [1] [2] [3] [4] [5] and [6]). To make the document self-contained, the required perquisites are provided herein.

2.1 Notation and Terminology

In what follows we use the following notations: instances of data are denoted $x_1, \dots, x_n$, and are commonly given such that each x_i is a vector in d dimensional Euclidean space $\mathbb{R}^d$. The sequence $x_1, \dots, x_n$ is also referred to as the sequence of samples, and may be divided into training set (used for determining a model for later use), test set and validation sets (used for evaluating the performance of the selected model). In some cases data samples are given along with pre-defined labels, or classes, usually from a closed set of possible such labels: $\{C_1, \dots, C_N\}$, in which case the input data takes the form $(x_1, y_1), \dots, (x_n, y_n)$, and each of the y_i's is an element of $\{C_1, \dots, C_N\}$. For example, a typical learning problem is the classification of a new data instance $x \in \mathbb{R}^d$, given the training data $(x_1, y_1), \dots, (x_n, y_n)$. In other words, assign a class/label y to the new instance x, using the training data to learn some model. A concrete realization of this example might be the following case: the instances of the sampled data $x_1, \dots, x_n$ are voice samples of several speakers from a list of N speakers, each sample is labeled with its correct speaker $y_i \in \{C_1, \dots, C_N\}$, and the problem is, given a new voice sample x, assign it a correct speaker or declare it as "unknown" (not a member of the closed set of given speakers). This specific example is further treated in Section 4.

The model that is used to for prediction/classification is typically learned from training data, and is aimed to generalize to new instances. A model that performs poorly on the test set while achieving good results on the training data, is said to be *overfitting*. Overfitting is a well-known challenge in learning problems, and is, roughly speaking, a symptom of a too complex model being fed with not rich enough data. The collection, manipulation, augmentation and correct use of large data sets is one of the main challenges in the field. In each of the presented problem domains in this document, we provide a brief review of well-known and publically available data sets (while some data sets we have used are the result of our own collection effort).

The model obtained by the learning process from the data samples, typically depends on a set of parameters $w = (w_1, \dots, w_k)$, and can take the form of a function of the data sample (new instance) x, further parameterized by w, hence denoted $f_w(x)$. The model f_w is sometimes used directly to assign a class y to the new instance x. In other cases, as presented in this report, it can be used to extract *feature vectors*, or *signatures*, that can further be processed to assign a class to the new instance. The term *feature vector* is frequently used in this report, and may be used interchangeably with the term *signature*. The feature vectors, or signatures, are representations of the data sample x that enable better characterization, discrimination, etc. of data samples. Hence, feature extraction is a common step in solving the problem. The feature vector may be of different dimension than the data sample, and is then said to be "embedded in feature space".

Learning (or training) in our context, is the process of finding a specific set of weights w, optimal in some sense, that determine a specific model from the family of possible models parameterized by w. Many optimality criteria exist for searching the set of weights, which are implemented as a scalar function to minimize, also referred to as the *cost, penalty, objective* or *energy* function. Thus, the learning process is closely related to mathematical optimization, as it is the minimization problem of a (typically computationally complex) cost function.

For completeness, we note that most of the above terminology refers to the type of learning called *supervised learning*. Specifically, whenever the data samples are provided with labels y_i, the process is referred to as supervised learning, since the labels act as supervisors, or "teachers". This need not always be the case, as the problem might be learning a pattern from the data samples only. The canonical example to explain the notion of *unsupervised learning* is the problem of clustering of a set of given points (see, for example, chapter 22 of [5]). The majority of the methods used to generate the results in this report belong to supervised learning methods. We have, however, used unsupervised learning (such as clustering methods) in several cases. Other classifications of learning paradigms exist (other than supervised vs. unsupervised) but are out of scope of this document.

2.2 Some Useful Classical Learning Algorithms

Prior to describing the deep learning methods, we provide a high level overview of classical methods. In each of our solutions, various algorithms are incorporated in pre-process and post-process, supporting several additional tasks apart from the deep neural network model. A (partial) selected list of relevant methods is described herein, for an extensive study see [5], [3] and more.

Linear models/predictors are commonly used and are popular since their computational complexity is fairly low. The learning process typically amounts to solving a linear system of equations or a linear optimization problem, and sometimes quadratic minimization with linear constraints. The assumed underlying geometry is that the data samples can be approximated (or separated, depending on the application) by lines, planes, half-spaces, etc. Although this need not always be the case, for many applications the linearity of the model proves useful. This class includes linear regression, logistic regression, and half-space separators. For example, in some problems, a multi-class logistic regression classifier successfully serves as the final stage of the algorithm, after some complicated and non-linear feature extraction took place.

The next family of useful methods worth mentioning belongs to the concept of feature extraction and dimensionality reduction. We make extensive use of methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), which amount to a linear dimensionality reduction that is optimal in some sense relevant to the features of interest. Mainly for visualization purposes, we make use of the t-SNE [7] method.

We have encountered, in several cases, the need to perform straight forward clustering of data points (feature vectors). We make use of the well-known k-means algorithm with some variants. Another useful tool is the simple k-nearest neighbor (KNN) algorithm. In this case, the underlying geometry assumed, suggests that in the embedding feature space, where a proper metric is selected, proximity in terms of the distance function implies proximity of the interesting features, or the required label/class. Some further standard mathematical and statistical are used, and are not the objective of this document. The main tool we use are the deep neural networks, described next.

2.3 Neural Networks

Given an input vector $x \in \mathbb{R}^d$, consider a matrix W of weights and a vector of biases b, let $y = Wx + b$ be the vector of (translated) weighted combinations of x. An activation function $\sigma: \mathbb{R} \to \mathbb{R}$ can then be applied (element-wise), yielding $a = \sigma(y) = \sigma(Ax + b)$. By repeating the process with the resulting vector a, using function composition, we obtain a family of functions of the input x. The family is parameterized by the matrices of weights and biases W's and b's, that can be modelled in layers of "neurons" (Figure 1), where each layer (or even an individual neuron) is equipped with its activation function. The links between the neurons are equipped with the weights, and the biases are modeled such they act as the weights of the constant neuron with output 1. This mathematical model is inspired by the structure of the neurons in the human brain, and has been investigated for several decades. However, in recent years, with the ability to perform complex computational tasks in tractable time, neural networks have been proven outstandingly useful in several application domains.

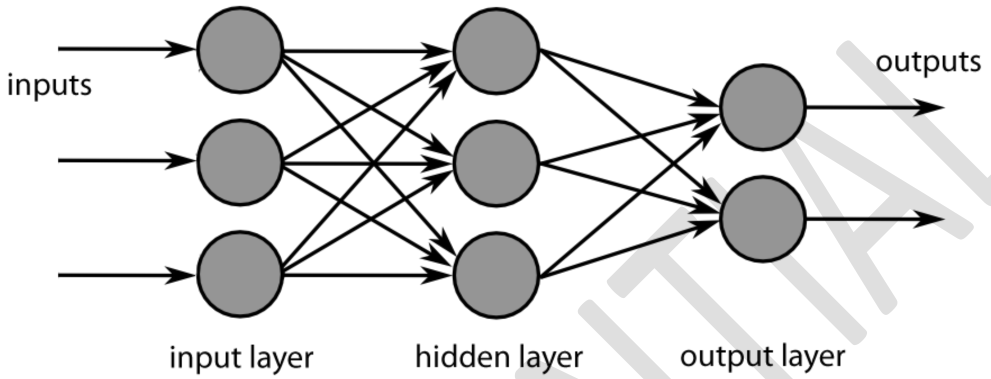


Figure 1: A neural network is a function of the input that can be organized in layers, parameterized by the weights and biases. Figure adopted from http://technobium.com/stock-market-prediction-using-neuroph-neural-networks/

Further degrees of freedom are enabled by the architecture and connectivity of the network, the selection of the specific activation function, and the selection of the cost function involved. Then, once minimized over a properly chosen data set of instances, the cost function is designed such that the weights and biases provide a powerful predictor/classifier/feature extractor (depending on the problem domain). As in all other models, the complexity of the neural network must be balanced with the size and complexity of the data set it operates on, or else the process may result in overfitting, for instance. This intuitive statement can be made precise in mathematical terms, and is out of scope. For an extensive study of the various types of neural networks and possible application, refer to [8], [9], [10]. The actual learning (minimization of the cost function) is typically performed by a stochastic version of the well-known gradient descent optimization algorithm. The actual evaluation of the gradient is referred to as *back-propagation*, due to the fact that differentiation of the composition of functions reflects, in the network model, as evaluating derivatives backwards along the layers. The evaluation of the actual network function on a specific input vector is referred to, naturally as *forward* evaluation.

2.3.1 Special and Useful Types of Neural Nets

The term *deep learning* refers to the process of training a neural network with hidden layers (namely, more than merely input and output layers. In the field of computer vision, *convolutional* neural networks have proven very useful, taking the role of manually crafted features classically performed in image processing methods, enabling the detection of local features to take place automatically by the learning procedure. The convolutional nature of the network architecture enables the detection of local features and has established and impressive results in several problem domains [11], [12], [13]. Other types of networks can handle time series and mimic models with memory, implemented by allowing cycles in the network topological structure. These are called recurrent neural networks (RNN) and are useful in various problem domains such as language translation, pattern recognition and prediction, and more [14], [15].

3 Computer Vision

This section provides a description of our efforts on selected computer vision problems. We report results on various problems in face recognition and object recognition domains.

3.1 Problem Definition and Solution Outline

The two major problems addressed are face recognition and object recognition. Face Recognition is a term that includes several sub-problems. There are different classifications of these problems in the literature. The input of a face recognition system is an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video. Our solution pipeline is outlined in Figure 2.

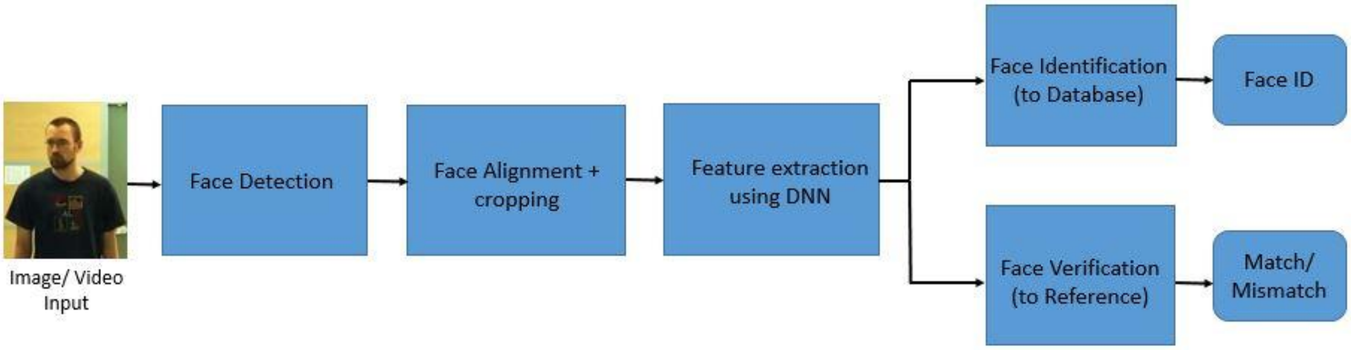

Image/ Video
Input
Face Detection
Face Alignment + cropping
Feature extraction using DNN
Face Identification (to Database)
Face ID
Face Verification (to Reference)
Match/ Mismatch

Figure 2: Face recognition, solution outline

Face detection is defined as the process of extracting faces from scenes. The system identifies a certain region in the image as a face. This procedure has many applications like face tracking, pose estimation or compression. The next step, feature extraction, involves obtaining relevant facial features from the data. These features could be certain face regions, variations, angles or measures. Feature extraction involves several steps - dimensionality reduction, feature extraction and feature selection. This steps may overlap, and dimensionality reduction could be seen as a consequence of the feature extraction and selection algorithms. Both algorithms could also be defined as cases of dimensionality reduction. Finally, the system does recognize the face. In an identification task, the system would report an identity from a database. This phase involves a comparison method, a classification algorithm and an accuracy measure. This phase uses methods common to many other areas which also do some classification process. Note that our operational scenario involves a set of known entities, where the query image (new instance) cannot be assumed to belong to them. In such case, we are required to assign it an "unknown" label.

As for object recognition, the problem is defined as assigning a class to an image from a closed set of possible classes of objects. Unlike the face recognition problem, we did not take on the challenge of detecting unknown objects (i.e. correctly recognizing them as not belonging to the closed set of classes).

3.2 Datasets

We describe several commonly used datasets in the problem domain of face and object recognition. First, LFW (Labeled Faces in the Wild) is a face photographs dataset for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector. This data set is the de-facto standard for researchers to benchmark and evaluate their face recognition methods. We report our results on the standard protocol, which differs from our operational use-case, as will be evident in the next sections. A larger dataset for face recognition is the CASIA-Web Face dataset. It contains 10,575 subjects and 494,414 images using semi-automatically way to collect face images from Internet. We report results with respect to CASIA, referring to our operational use-case. LFW and CASIA-Web Face are the data sets we focus on in this report (several other datasets for face recognition are available and commonly used, and are not part of our current effort).

For object recognition we concentrate on the CIFAR10 dataset and the ImageNet dataset. CIFAR-10 contains 60,000 32x32 color images in ten classes, with 6000 images per class. There are 50,000 training images and 10,000 test images in the official data. The classes are mutually exclusive. ImageNet is an image dataset organized according to class hierarchy. ImageNet aims to provide on average 1000 images to illustrate each class/concept ("synset"). Images of each "concept" are quality-controlled and human-annotated. Other well-known datasets for object recognition exist, and are not the focus of this report.

3.3 Related Work and Benchmarks

This section provides a literature review of face recognition and object classification results. Before Deep Neural Networks arose, the conventional machine-learning techniques were limited in their ability to process natural data in their raw form [8]. In general, those techniques were based on designing a feature extractor that transformed the raw data into a suitable feature vector from which the learning subsystem could detect or classify patterns in the input. For decades, researchers had supplied constant but very slow improvement path in results, when fundamental limitations of these hand-crafted feature engineering approaches could be summarized as:

1. Lack of ability to process data in its natural form
2. Considerable domain expertise required to design feature extractors; time-consuming for data scientists and SW engineers
3. Highly specialized methods; lack of generalization
4. Results were non-comparable with human performance. For example, in the ImageNet large-scale visual recognition challenge (World Cup for computer vision and machine learning) in 2011, 2.5% error rate for humans vs. 26% error rate for state-of-the art non-DNN solution.

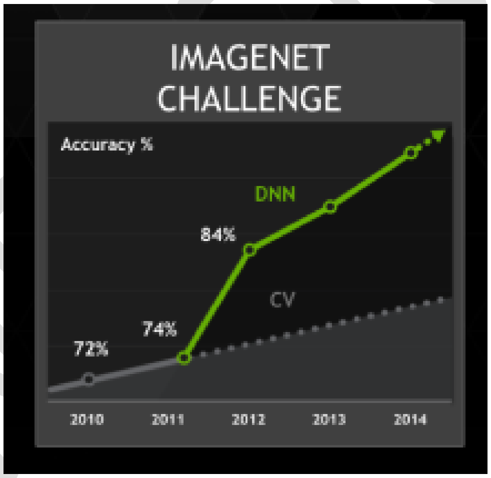


Figure 3: DNN revolution on ImageNet challenge example

Deep learning has produced extremely promising results for various visual recognition tasks. As depicted in Figure 3, in 2012, deep convolutional networks were applied to a data set of about a million images from the web that contained 1,000 different classes, achieving spectacular results, almost halving the error rates of the best competing approaches [16].

ConvNet architectures. Deep learning has brought about a revolution in computer vision. Convolutional neural network (ConvNets) are now the dominant approach for almost all recognition and detection tasks, and approach human performance on some tasks ([8], [17],

[18], [12], [19]). ConvNet architectures make the explicit assumption that the inputs are images, enabling to encode certain properties into the architecture. These then make the forward function more efficient to implement, and significantly reduces the amount of parameters in the network [20]. Main components of ConvNet success came from advances in GPU training, regularization technique called "dropout" [21], and data augmentation techniques - generating more training examples by deforming the existing ones.

Three main types of layers compose the ConvNet architectures: Convolutional Layers, Pooling Layers, and Fully-Connected Layers. All ConvNet architectures combine these layers to form a full ConvNet architecture. Recent ConvNet architectures have 10 to 150 layers, hundreds of millions of weights, and billions of connections between units. Further ConvNet architectures are the VGGNet architecture [12] and the recent ResNet, developed by Microsoft [22], winning ILSVRC 2015.

Novelty detection. A good measure of similarity allows us to not only classify objects using similar objects, but also detect the arrival of a new class of objects. For a review of the topic see [23]. As will be evident from the remainder of this section, deep convolutional neural networks, similarity measures and novelty detection methods are at the core of our solution.

3.4 Our Experiments

We report our results on several face recognition tasks, with respect to standard academic benchmark protocols, as well as with respect to our own evaluation method, as derived from our anticipated use-case scenario.

3.4.1 The Academic Benchmark

In order to allow for direct comparison to previous work, evaluation is performed on an existing benchmark dataset. As mentioned above, the chosen benchmark dataset is Labeled Faces in the Wild dataset (LFW) [24]. It contains 13,233 images with 5,749 identities, and is the standard benchmark for automatic face verification. For evaluation, LFW is divided into predefined splits for 10-fold cross validation. Each time nine of them are used for model training and the other one (600 image pairs) for testing. LFW defines three standard protocols (unsupervised, restricted and unrestricted) to evaluate face recognition performance. "Unrestricted" protocol is applied here because the information of both subject identities and matched/unmatched labels is used in our system. The face recognition rate is evaluated by mean classification accuracy and standard error of the mean. The LFW images we used are aligned by deep funneling [25], then cropped and

resized according to our needs. We have compared our results to several known academic papers (Table 1).

No.	Networks	Accuracy
Fisher Vector Faces [26]	-	93.10
DeepFace [18] (Facebook)	3	97.35
VGGNet [12] (Authors currently at Google)	1	98.95
Fifth Dimension	1	96.52

Table 1: Results on the standard LFW benchmark protocol of various methods and ours

3.4.2 Our Operational Use-Case

Our operational use case is slightly different from the verification benchmark. We still use a DNN for feature extraction (signature), but the problem of our interest is the ability to recognize the face in the new query image, from a set of many known entities, or declare it as "unknown" (i.e. not in the closed set of known entities). As for the detection of unknowns, we have learned a threshold in terms of our similarity function in feature space. This process, however, is performed in a similar manner to the one detailed in Section 4.5.2.3, for speaker recognition, and hence not described herein.

To evaluate our performance, we either traverse the entire data set, taking each single image as the query against the rest, in turn, or divide our data sets into disjoint sets of codebook and test sets. The reason for this difference, is that one of our tested options required further optimizing the feature vectors, via a mechanism that uses a part of the data set. To create a fair and valid test, we must separate the possible queries from the data against which it tested, since all the feature vector optimization cannot be performed on data that later serves is the query/test data. The DNN is used for feature extraction, and the parameters for the signatures similarity are computed on different data. Both the codebook and the test set were not unknown to the system. First, we report results on experiments in which the codebook contains all the entities in the test set (namely, no unknowns). Results reflecting our unknown entity detection capability are presented in Section 3.4.3. We present the accuracy in the following sense: the system returns the five most similar entities, and success is considered an event where the correct entity is within the five most similar ones. For this criteria, we report 94.0% on the CASIA dataset, and 96.1% on the LFW dataset.

3.4.3 Further Results – Surveillance Camera Images and Object Recognition

Security surveillance systems often produce poor-quality video, and this may be problematic in gathering forensic evidence. We examined our ability to sustain our results previously reported on faces captured by a commercially available video security device.

SCface dataset [27] was designed mainly as a means of testing face recognition algorithms in real-world conditions. In such a setup, one can easily imagine a scenario where an individual should be recognized comparing one frontal mug shot image to a low quality video surveillance still image. In order to achieve a realistic setup, images are taken with commercially available surveillance cameras of varying quality. The dataset contains also different head pose images. Since two of the surveillance cameras record both visible spectrum and IR night vision images, IR imagery is included in the dataset as well.

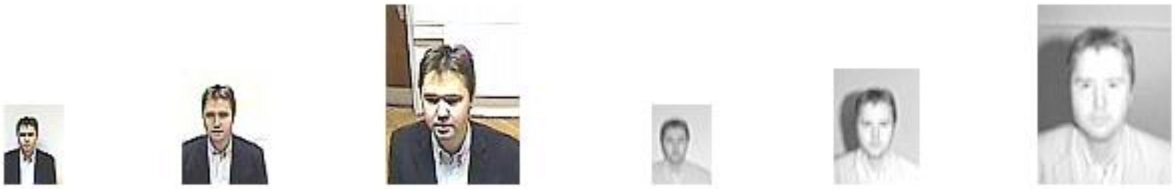


Figure 4: SCFace – visible light and IR frontal mug shots with different quality and resolution

We have tested our system on two different subsets of the SCface dataset. The first was a very challenging one with the full dataset including IR and head pose images and the second consist of a subset of the full data set according to Face Authentication Protocol [27], [28].

In our full challenge, we used 2209 images from 120 (out of 130) subjects, hence we were required to detect unknowns as well as correctly recognize known subjects. In this use-case, we have achieved a score of 86.9% accuracy for a single returned entity, and 90.0% accuracy for three returned entities (success considered as correct identity within the three). When tested without the need to detect unknown entities (namely a classification problem from a closed set) our results were 93.7% for a single returned entity, and 98.4% accuracy for three returned entities.

In the second protocol we have chosen a subset of daylight mugshots with various resolutions and qualities (as proposed by the creators of the data set). In this challenge, we have achieved a score of 95.7% accuracy for a single returned entity.

As for object recognition – on the CIFAR-10 dataset our accuracy result is 94.21% for correctly classifying the given objects, whereas the state of the art results are 95.6% (see, for example [29], [30]). On the ImageNet dataset, our top-5 accuracy results are 93.7%, whereas the state of the art result is 94.3% for top-5 accuracy [22] (Microsoft).

4 Speaker Recognition

Speech, apart from being the most common ways of human communication, is unique and discriminative – carrying the identity of the speaker as voice print (like fingerprints). Human speech can be represented as a signal containing various types of information: words, sentiment, language and identity of the speaker. In several applications, the capability to correctly recognize a persons is required. The use of biometric-based (physiological and/or behavioral characteristics of a person) recognition is a natural approach to recognizing a person, and is considered safe, as these characteristics are difficult to steal or forget.

Automatic Speaker Recognition (ASR) is the problem domain addressing the recognition of a person based on his/her voice. Speaker recognition systems can be classified as speaker identification or speaker verification systems:

- Speaker Identification is the task of finding who is talking from a set of known voices of speakers. The unknown voice comes from a fixed set of known speakers, hence the problem is also referred to as closed set identification. Speaker identification is a 1: N match where the voice is compared against N templates. Error that can occur in speaker identification is the false identification of speaker

- Speaker Verification is the binary classification problem (true/false) with respect to a speaker claiming to be the actual one. It is assumed that the other speakers are not known to the system, and this problem is also referred to as the open set task. Speaker verification is a 1:1 match where one speaker's voice is matched to one template

Two additional problems of our interest are speaker detection and speaker separation. Speaker detection is the process of making a decision whether the target speaker is present in an audio stream involving various speakers. This is similar to speaker verification, however instead of comparing a speech utterance of single speaker, we compare a whole stream with the reference speaker models. The speaker separation (audio diarization) is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. Speaker separation is a combination of speaker segmentation and speaker clustering. The first aims at finding speaker change points in an audio stream. The second aims at grouping together speech segments on the basis of speaker characteristics.

4.1 Problem Definition

The problem we aim to solve is the speaker identification in a close set of known speakers, with the additional need to detect unknown speakers (label them as "not in the set"). Our system has two main flows: Initialization and Processing. During the initialization phase, a set of tagged audio samples are been injected into system, each sample is pre-processed and a voice print is extracted using our model and associated with the known speaker. In the processing flow (i.e. the speaker recognition), a new voice sample is been processed for voice print extraction using the same model, and then identifying the speaker by comparing the voice print of the processed voice sample against the voice print of the known voice samples as were extracted in the initialization flow.

4.2 Datasets

We report results on two commonly used datasets, TIMIT and HYKE. TIMIT is a corpus of phonemically and lexically transcribed speech of American English speakers of different genders and dialects. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions, as well as a 16-bit, 16kHz speech waveform file for each utterance. The HYKE dataset was collected for the purpose of speaker identification in developing country contexts. It includes a total of 83 unique voices, 35 female and 48 male. In particular, it provides audio for performing limited vocabulary speaker identification using digit utterances. The data was collected in partnership with Microsoft Research India. The data was collected over the telephone using an IVR (Interactive voice response) system in March of 2011 in India. The participants are Indian nationals from various backgrounds. Each participant was given a few lines of digits, and asked to read the numbers after getting prompted in the system. Each participant read five lines of digits, one digit at a time. The numbers were all read in English. There is various levels of background noise, ranging from faint hisses to audible conversations or songs. In total, about 30% of the audio has some level of background noise.

4.3 Related Work

The speaker recognition task has been vastly researched within the context of biometric systems for recognition. Classical approaches are based on probabilistic models as GMM (Gaussian Mixture Model) and UBM (Universal Background Model) [31] of the speakers, where a model is learned for each speaker. The model corresponds to features that were extracted from the given samples (training data) of this specific speaker. In this method, features that are extracted from the samples of a single speaker are fit to a set of Gaussians (or other probabilistic model) in the feature space (typically high dimensional).

To create the voice print of the speaker, a set of features is extracted from the audio samples. The features are aimed to represent a voice print of the speaker, and used for training a model. Researchers try to understand the process of producing the human voice using the vocal apparatus, and the extracted features are related to its structure. A common method to extract those features is by applying signal processing techniques such as Short-Time Fourier transform (STFT) and spectrograms, which converts the voice amplitude to its representation in the frequency domain. Another technique to extract the set of features is done by using the Mel-Frequency Cepstral Coefficients (MFCCs) [32] which approximates the human auditory system's response. The main problem using those techniques is when the audio sample was recorder in noisy environment. In [33], the presented method accounts both for the speaker and the environment "channels" in the voice sample, and converts it to one representation called i-vector.

In recent years, as the amount of data grows and the computational capabilities of modern systems increases, the use of Deep Neural Networks (DNN) as multilayer perceptron (MLP) [34, 9] or convolutional (CNN) [13] also increased, yielding better results. In these approaches, based on DNN, the feature extraction is done by the neural network as part of the enrollment phase, then some output of the neural network is used as the new features for the identification model. As described next, our approach also belongs to this class of methods.

4.4 Overview of Our Approach

As described above, the problem of our interest is the ability to recognize the speaker in an audio signal. This task is also known as speaker identification. We assume a codebook is given, containing a certain amount of speakers. Given a new audio file we are interested in recognizing whether the speaker is unknown (i.e., not one of the speakers in our codebook), or a known speaker and we return the top N (typically, 5) speakers and a certainty score between 0 and 100.

Our speaker recognition system comprises three parts (Figure 5):

1. Preprocessing of the audio signal
2. Extracting signatures (feature vectors) using a deep neural network (DNN)

3. Identifying the speaker based on signatures similarity.

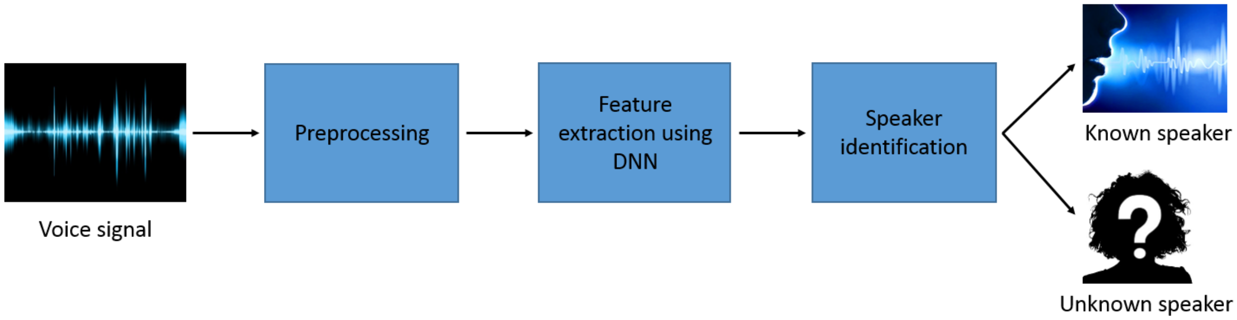


Figure 5: Speaker recognition algorithm flow

Unlike most of the speaker recognition works that can be found in the literature, in our system we are facing uncontrolled audio signals: multiple speakers, different sound qualities and capturing the audio signal with various modalities. In our preprocessing step we address some of these challenges. For example, we detect the number of speakers in the audio file, and split it to different audio signals that contains one speaker only. In the preprocessing step we also handle different sound qualities by using simple de-noising algorithms. Following the preprocessing step, we compute a signature for every audio signal. The signature is computed using a deep neural net that we have trained. Using the signature (also referred to as the feature vector), we can detect whether the speaker is unknown, or we can find the closest N entities based on some similarity measure in the feature space.

Speaker identification can be addressed as a classification task. However, since our use-case may include datasets with a large number of speakers, we decided to compute the features using DNN and then identify the speaker by signatures similarity (rather than a classification approach on the features, from a closed set). We are able to detect whether the speaker is unknown (i.e., not one of the speakers in our codebook), if the speaker is known we also return our certainty score, based on the similarity measure.

4.5 Our experiments

In all of our experiments we split the data into codebook and test set. The DNN used for feature extraction, and the parameters for the signatures similarity are computed on different data, both the codebook and the test set were not seen before – to test our method in as close as possible conditions to our forecasted scenario. We report results on experiments in which the codebook contains all the speakers in the test set (namely, no unknowns), in terms of the accuracy. If we return more than one speaker, we consider success if one of the returned speakers is the true speaker.

4.5.1 The Academic Benchmark

To evaluate our speaker recognition system, we followed the protocol described in [35]. The results are reported for the TIMIT speech dataset, which contains audio samples from 630 speakers and each speaker have 10 audio samples. TIMIT does not have a standard decomposition into training, validation and test sets which is suitable for work with speaker recognition.

In [35], the authors divided the 630 speakers into disjoint training, validation and test sets of 300, 162 and 168 speakers receptively. The test set is divided into disjoint codebook and test sets, by choosing, at random, for each speaker, a certain amount of samples for the codebook. The test set does not include unknown speakers. The authors report results with different models, best result achieves very high accuracy of 98.7%.

To evaluate and compare ourselves, we have made an effort to follow the protocol in [35] as closely as possible. Since not entirely explicit and clear, we changed the protocol by dividing the 630 speakers into disjoint training and test sets of 100 and 530 speakers receptively (the specific test set was not stated in [35], hence we randomized the selection process). The training set was further divided into disjoint training and validation sets, and the test set was further divided into disjoint codebook and test sets.

We report the accuracy results for returning N=1, 3, 5 speakers, where success is considered when the correct speaker is one of the returned ones. We also report results for testing our speaker recognition system on 168 speakers. In this case, we randomly chose 168 out of the 530 speakers that were originally selected for our large test set, and compute the accuracy on this small test set. We repeat this process; in each repetition we choose a different set of 168 speakers for the test set. We compute the average accuracy on these tests and achieve comparable results to the ones presented in [35] (

Table 2).

Model	**Result for N = 1**
MLP_d [35] – on 168 speakers	98.7%
Fifth Dimension – average accuracy on 168 speakers	97.2%

Table 2: Accuracy results on the academic benchmark on TIMIT speech dataset

4.5.2 Our Operational Use-Case

We have tested our system on two public datasets, TIMIT and HYKE, and on additional data that was collected in our company. We present the accuracy results with N=1, 3, 5 top speakers. In order to evaluate our signature extraction process, we use t-SNE [7]. t-SNE, t-distributed Stochastic Neighbor Embedding, is a technique used for nonlinear dimensionality reduction. Roughly speaking, it models each high-dimensional vector by a two or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points. Figure 6 shows the signatures for 15 speakers from TIMIT database, after t-SNE dimensionality reduction. It illustrates how the extracted signatures using our DNN are well separated across different speakers.

4.5.2.1 Experiments on HYKE

As mentioned above, the speakers in HYKE all read in English, there is various levels of background noise, ranging from faint background conversation to audible songs. About 30% of the audio has some level of background noise.

In our experiments we used a codebook containing audio signals of all the speakers (83) and a test set that contains different audio files of the same speakers each one with three examples. The total size of the test set contains 249 audio samples. Our system was able to recognize the true speaker with very high accuracy scores although some of the audio has background noise. For our operational case, returning 5 speakers, we achieved very high accuracy of 98.4% (Table 3).

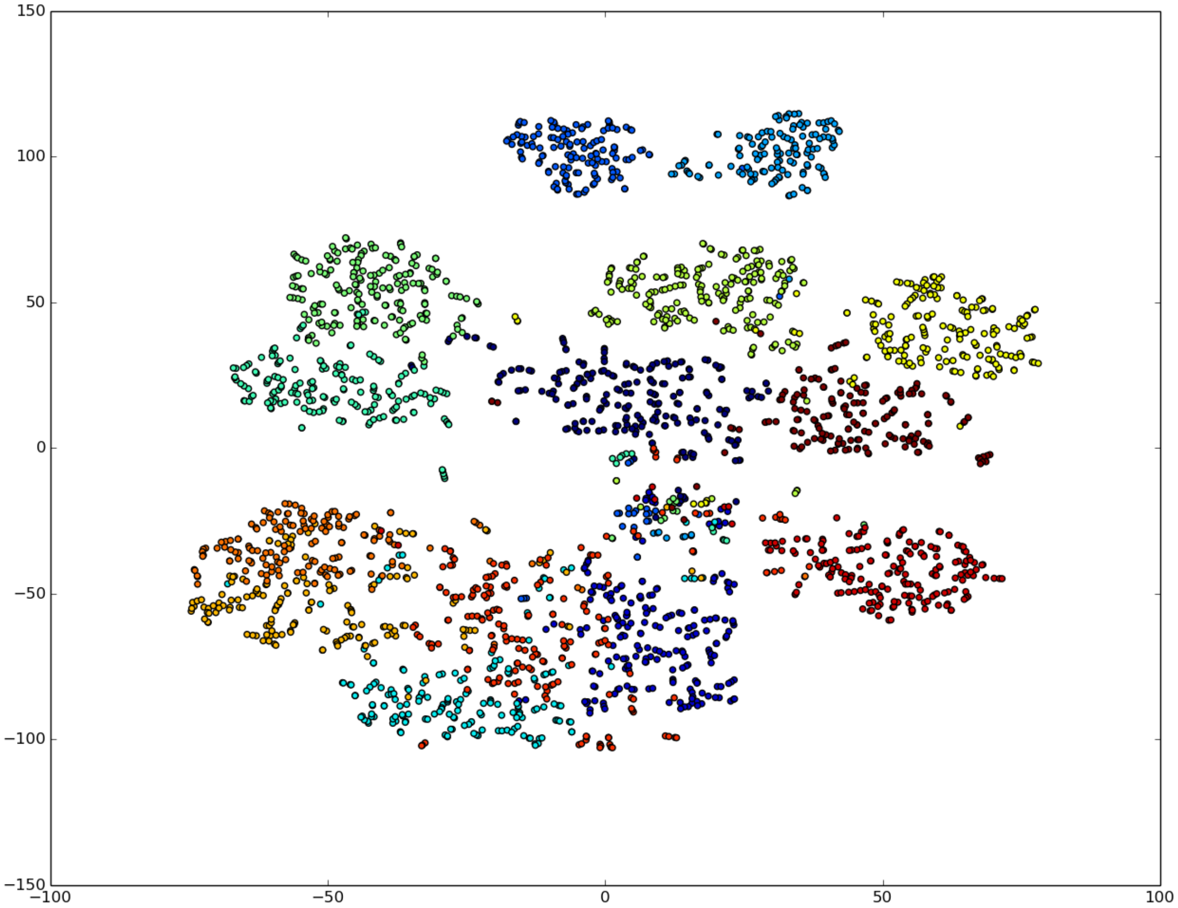


Figure 6: Signatures of 15 different speakers from TIMIT database, after dimensionality reduction using t-SNE. Colors are given according to the speakers' true identity.

4.5.2.2 Experiments on TIMIT

In a similar manner to the academic benchmark, we divided the Timit speech dataset which contains 630 speakers, into disjoint training and test sets of 100 and 530 speakers respectively. The DNN for signatures extraction and the parameters for the signatures similarity, were trained on the training set and on additional audio samples that were collected in our company. We report the accuracy results with N = 1, 3, 5 speakers on the test set containing 530 speakers.

For returning N = 5 speakers (our operational case) we were able to achieve high accuracy 97.5%. This result is comparable to the results we achieved on the test we performed on HYKE dataset, however in this protocol, the test set was much larger. Table 3 summarize our results on TIMIT and Hyke datasets.

Dataset	**N = 1**	**N = 3**	**N = 5**
TIMIT	90.6%	96%	97.5%
Hyke	93.6%	98%	98.4%

Table 3: Accuracy results on TIMIT and Hyke

4.5.2.3 Detecting Unknowns

Our operational use case also requires the ability to detect whether the speaker in an audio signal is unknown, namely, not one of the speakers in our dataset. We extended the signature similarity step in our system to support this requirement. When returning the top N most similar speakers for audio signal we compute a similarity score. We define a threshold on the similarity score, such that if none of the N speaker returned is larger than the threshold then we define the audio signal as "unknown" speaker.

Detecting unknowns can be referred as a binary classification task with the following requirements:

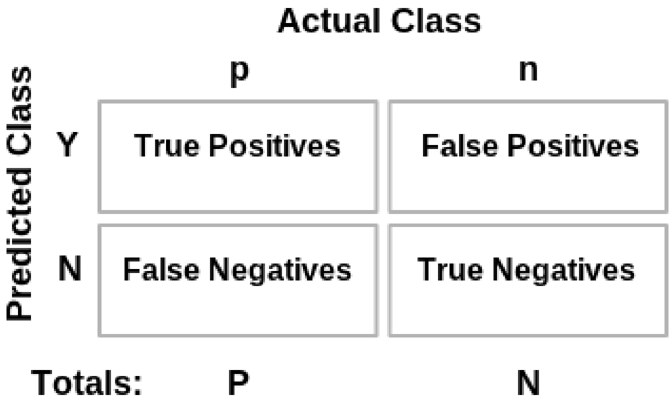
1. High precision – when our algorithm determine that the audio signal belongs to an unknown speaker, there will be almost no mistakes. Namely, the number of times we classify a speech signal of a known speaker as "unknown" will be as small as possible.
2. High recall – our algorithm is required to classify as many as possible audio signals of unknown speakers as "unknown".

Precision and recall are defined as follows:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives},$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Where the definition of the sets True Positives, False Positives and False Negatives are presented in the confusion matrix in Figure 7.



Actual Class

Predicted Class	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives
Totals:	P	N

Figure 7: Confusion matrix of a binary classifier

Unfortunately, in most cases increasing the recall, which, in this context, interprets to detecting more samples of unknown speakers as "unknown", results in decreasing the precision, since more samples of known speakers are also detected as "unknown". This is a well-known and inherent tradeoff.

To find the threshold meeting the precision requirements while maximizing the recall, we perform tests on TMIT dataset (the method is applicable to any data set). After training our speaker recognition system, we divide the test set which contains audio samples of 530 speakers as described in our test protocol into disjoint codebook and a new test set as follows:

1. We randomly select 15% different speakers (in our case 80 speakers) and add all their samples to the new test set.
2. For every speaker that wasn't selected in step 1, we choose at random a certain amount of samples for the codebook and add the rest to the new test set.

The new test set contains both samples of known and unknown speakers, in our case the samples belongs to unknown speakers are about 20% of the entire test set. For every sample in the test, we run our system and return the most similar speaker and a similarity score. We repeat the same test a few times (typically, 10 times), and evaluate the probability of score values, given the unknown/known status. We denote these probabilities by $p(query|known)$ and $p(query|unknown)$ respectively.

Using these probability estimation, we can determine a threshold on the similarity score. We choose the threshold according to the above mentioned requirements. Figure 8 illustrates the estimated probabilities, $p(query|known)$ and $p(query|unknown)$. One can see that if we decrease the threshold, the precision will increase (less of samples of known speakers will be classified as "unknown"), on the other hand the recall will decrease.

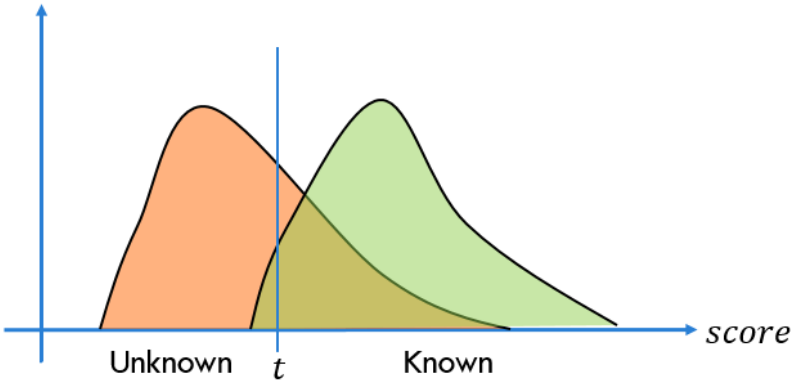


Figure 8: Illustration of the estimated probabilities for scores for known and unknown speakers

5 Text Similarity

Classifying text documents into certain categories (or associating them with a certain topic) is a common problem in the domain of NLP (Natural Language Processing). One possible approach is the extraction of entities and specific terminology from the text. Over the last decade, methods for creating semantic representations for entire documents have been developed. Such representations are, in turn, compared to the semantic representations of existing documents belonging to categories or topics. The topic or category of the most similar existing semantic representation will be assigned to the new text document. This process is akin to human understanding of text, for example: consider a person reading an unnamed document and filing it in a certain folder, based on its content. Such text recognition is based not only on the words contained in the document and on their frequencies, but also on the context and the meaning of the words, sentences and paragraphs in the text. Scenarios of this nature motivate our use-case and methods, described herein.

5.1 Problem Definition and Solution Outline

In classifying text based on document similarity, we address two problems:

1. **Creating a representation of the text**.

   The representation transformation must be able to accept as input, a document of any size (a document may be comprised of a few words or many pages of paragraphs), and is required to output a representation which is a feature vector of a fixed, predefined dimension.

   The representation problem is currently addressed via two main algorithms: **Bag-of-Words** and **Paragraph Vectors** (commonly referred to as doc2vec). In the popular (and simpler) Bag-of-Words model (for a review, see [36]), the (frequency of) occurrence of each word is used as a feature in the final representation. While useful for many applications, this model has two major drawbacks: the ordering of the words is lost, and semantics of words is ignored. For example, the words "security," "safety" and "Paris" are equally distant. Paragraph Vectors [37], on the other hand, is an unsupervised neural network algorithm that overcomes both weaknesses. Moreover, it is more efficient in terms of memory consumption. In this algorithm, each document is represented by a dense vector, which is trained to predict words in the document. The study by Le and Mikolov [37] shows that the Paragraph Vector model outperforms Bag-of-Words models, as well as other techniques for text representations.
2. **Finding a similar document.**

   Once a new document arrives into the pipeline, it must be assigned a topic from one of the topics existing in the stored data. We consider two classes of possible solutions. One approach is to train some supervised learning classifier using all the text representations already stored, and classify the new document according to its representation. Another option is comparing the representation of the new document to the stored representations of the existing documents, using some similarity function. The similarity function compares two representations and returns a score based on how similar they are to each other – as viewed by the chosen metric in the space of representation vectors. The similarity results can then be used either to label the new document, or suggest similar documents to the user. Note that a common situation in recommendation systems is the learning of user choices, while our model does not (currently) rely on user choices, but on the feature similarity of the instances.

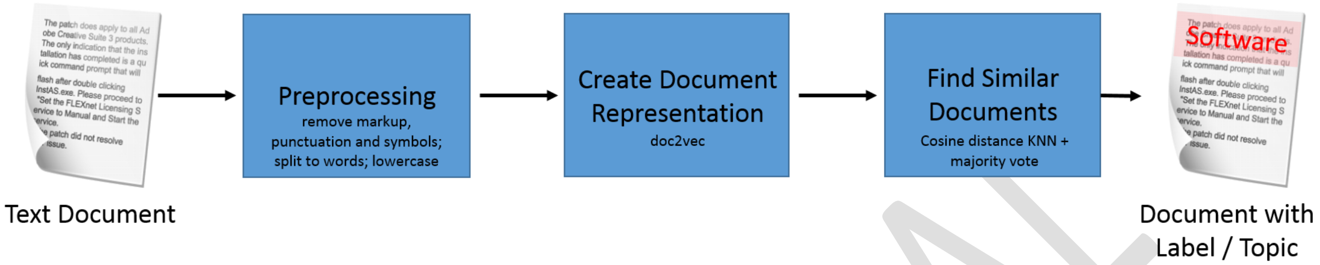

Text Document
Preprocessing
remove markup, punctuation and symbols; split to words; lowercase
Create Document Representation
doc2vec
Find Similar Documents
Cosine distance KNN + majority vote
Software
Document with Label / Topic

Figure 9. Our solution pipeline

5.2 Datasets

We describe several publically available datasets (text corpuses) used for testing our models:

1. **20 Newsgroups** (assembled by Ken Lang). The 20 Newsgroups data set is a collection of approximately 20,000 real-world newsgroup documents from the 1990s, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g. misc.forsale / soc.religion.christian). Below is a list of the 20 newsgroups, roughly partitioned according to subject matter:

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

The articles are typical postings and thus have headers including subject lines, signature files, and quoted portions of other articles. An example posting in sci.space:

```
Original to: wats@scicom.AlphaCDC.COM
G'day wats@scicom.AlphaCDC.COM

20 Apr 93 18:17, wats@scicom.AlphaCDC.COM wrote to All:

wAC> wats@scicom.AlphaCDC.COM (Bruce Watson), via Kralizec 3:713/602

wAC> The Apollo program cost something like $25 billion at a time when
wAC> the value of a dollar was worth more than it is now. No one would
wAC> take the offer.

If we assume 6% inflation since 1969, that $25B would be worth about $100B
GD reckon a moon mission today could cost only $10B. Thats a factor of ten
reduction in cost. It might be possible to reduce that number futher by
using a few shortcuts ( Russian rockets?).   Asuming it gets built, I think
the Delta Clipper could very well achive the goal.

ta

Ralph

--- GoldED 2.41+
* Origin: VULCAN'S WORLD - Sydney Australia (02) 635-1204  3:713/6
(3:713/635)
```

2. **IMDB – Large Movie Review Dataset** The IMDB dataset was first proposed by Maas et al. [38] as a benchmark for sentiment analysis. The 100,000 movie reviews are divided into three datasets: 25,000 labeled training instances, 25,000 labeled test instances and 50,000 unlabeled training instances, all taken from IMDB (the Internet Movie Database, http://imdb.com). There are two types of labels: Positive and Negative, which are balanced in both the training and the test set. Each movie review is made up of one sentence or more, and could contain HTML markup. An example of a negative review from the dataset:

```
I usually don't comment anything (i read the others opinions)... but this, this one I _have_
to comment... I was convinced do watch this movie by worlds like action, F-117 and other hi-
tech stuff, but by only few first minutes and I changed my mind... Lousy acting, lousy script
and a big science fiction.<br /><br />It's one of the worst movies I have ever seen...<br
/><br />Simply... don't bother...<br /><br />And one more thing, before any movie I usually
check user comments and rating on this site... 3.7 points and I give this movie a try, now I'm
wondering WHO rate this movie by giving it more than 2 points ??????????
```

3. **Reuters-21578** One of the most widely used test collection for text categorization research. The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987, and were originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system. It is much smaller than, and predates, the Reuters-RCV1 collection discussed in the next item below. There are multiple categories, the categories are overlapping

and non-exhaustive, and there are relationships among the categories. For some of our applications we used a subset of this dataset, taking only those documents that had one category only, leaving 65 categories and 9160 documents.

5.3 Related Work and Benchmarks

In the NLP field of text categorization, the most popular model has been the Bag of Words model and its extensions (for a review, see [36]). Many algorithms have recently been introduced to model phrases and paragraphs, usually based on representations of words and auto-encoders ([38], [39], [40], [41], [42], [43], [44]). Notably, the emergence of models with distributed representations for words (Word-Vector models), which were successful especially for statistical language modeling (c.f. [45], [46]), led to the extension of the model by Le & Mikilov [37], to a model named Paragraph-Vectors, a distributed representation of phrases and documents. Le & Mikolov measured their model's performance against current state of the art models (extended Bag of Words & LDA, as described in [38]; Multinomial Naive Bayes as described in [47] and Support Vector Machines as described in [48]). Their results (shown in Table 4), give a clear advantage to Paragraph Vectors over all other models. Distributed representations of phrases were also suggested by Socher et al. ([49], [50]), but their model is supervised and is not yet extended beyond single sentences.

5.4 Our Experiments

We describe our results with respect to academic benchmarks as well as with respect to our own evaluation protocol, derived from our use-case scenario.

5.4.1 The Academic Benchmark

Le & Mikolov, who proposed the Paragraph Vectors model [37], published a comparison of their model with current state of the art text-classification models on the IMDB dataset described in section 5.2. The task was to classify the test reviews as positive or negative reviews. After training a model to create document representations, a classifier was used (not elaborated on in the paper) yielding the output class (negative/positive) that was used for evaluation. The results surveyed in [37] show a clear advantage to the Paragraph Vector model, hence we have based our solution on it as well. In Table 4, we extracted only the leading results for each of the models surveyed in [37], along with our results.

5.4.2 Our Results on the Academic Protocol and our Operational Use-case

We used a version of the Paragraph Vector model to obtain text representations and tested it on several datasets. We evaluate our performance using the topic/category of the text as the true label, and measure the success events in the sense that the similar document returned has the same label. Results are depicted in Table 4. For the IMDB dataset we compare our results to the above mentioned academic benchmarks, while for the 20 Newsgroups dataset we report our own evaluation method, as derived from our use-case scenario (percent of correctly classified documents w.r.t their true topics/categories).

Dataset	Size (train/test)	Number of Labels	Method/author	Accuracy
IMDB	75000/25000	2	Paragraph Vector [37] (Author currently at Facebook, Google Brain in the past)	92.58%
IMDB	75000/25000	2	Fifth Dimension	89.3%
IMDB	75000/25000	2	Mass et al. [38] (Author currently at Baidu, Google Brain in the past)	88.89%
IMDB	75000/25000	2	Dahl et al. [48]	89.23%
IMDB	75000/25000	2	Wang et al. [47]	91.22%
20 Newsgroups	11314/7532	20	Fifth Dimension	90.3%
Reuters-21578	6577/2583	65	Fifth Dimension	89.6%

Table 4: Results, in percentage of the correct label of the returned document, with respect to the category of the new query document.

The following figures depict document representations in 2D space, for several datasets we processed. Each document is represented by a vector, which, in turn, is dimension reduced using the t-SNE [7] method for visualization. The different colors represent different categories.

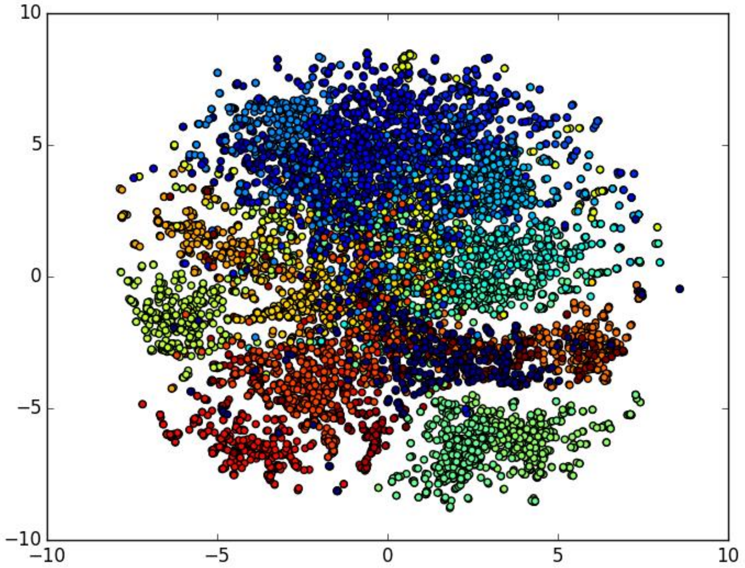


Figure 10: The 20 Newsgroups document representations in 2D space. Note, for example, the dark blue, maroon and orange overlapping categories on the bottom right quadrant are talk.religion.misc, alt.atheism and soc.religion.christian. Similarly, the blue and light blue categories on the top are computer related (comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x). Adjacent categories appear close to each other (e.g. the adjacent light green categories on the bottom right quadrant are rec.sport.baseball and rec.sport.hockey).

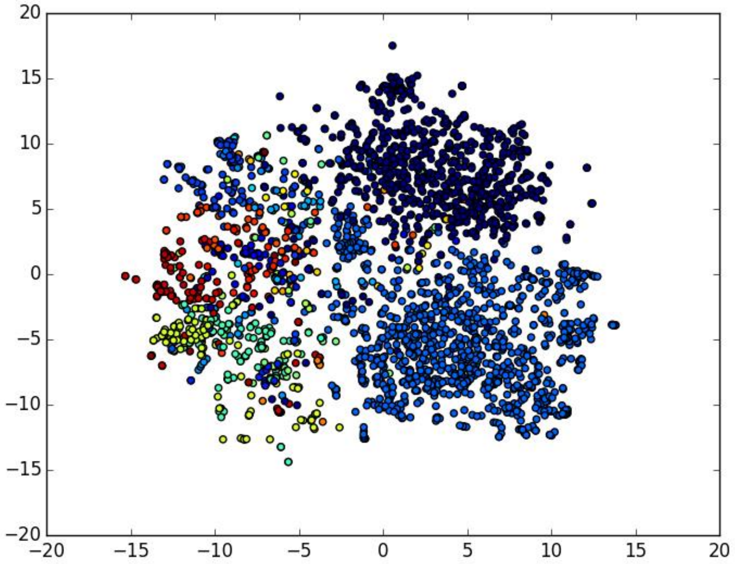


Figure 11: Reuters-21578 document representations in 2D space.

6 Conclusion

In this report, we have provided our results on selected problem domains addressed by the Fifth Dimension Research & Development group. We have focused on reporting quantifiable results, comparing ourselves to standard benchmarks where possible. Our solutions were outlined, with an attempt to provide a document that is self-contained with the required background material. Since our forecasted operational use-case is slightly different from the standard protocols, we have constructed own evaluation methods and added them to this report.

7 References

[1] Y. Abu-Mostafa, M. Magdon-Ismail and H.-T. Lin, Learning From Data, AMLBook, 2012.

[2] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

[3] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.

[4] K. P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.

[5] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[6] L. Wasserman, All of Statistics: A Concise Course in Statistical Inference, Springer , 2004.

[7] L. a. H. G. van der Maaten, "Visualizing High-Dimensional Data Using t-SNE," in *Journal of Machine Learning Research 9: 2579–2605*, 2008.

[8] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[9] D. R. Fred Richardson, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE SIGNAL PROCESSING LETTERS,* vol. 22, no. 10, pp. 1671-1675, 2015.

[10] R. Rojas, Neural Networks: A Systematic Introduction, Springer, 1996.

[11] M. . D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Computer vision–ECCV 2014,* 2014.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR,* vol. abs/1409.1556, 2014.

[13] M. McLaren, "Application of Convolutional Neural Networks to Speaker Recognition in Noisy Conditions," 2014.

[14] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," *arXiv:1503.04069,* 2015.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, p. 1735–1780, 1997.

[16] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Proc. Advances in Neural Information Processing Systems,* vol. 25, pp. 1090-1098, 2012.

[17] C. Szegedy, W. Liu and etc, "Going deeper with convolutions," *CVPR,* pp. 1-9, 2015.

[18] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *CVPR,* pp. 1701-1708, 2014.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *Proc. International Conference on Learning Representations,* 2013.

[20] A. Karpathy , "Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition," 2015.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *http://arxiv.org/abs/1512.03385,* 2015.

[23] M. Pimentel, D. Clifton, L. Clifton and L. Tarassenko, "A review of novelty detection," *Signal Processing,* vol. 99, p. 215–249, 2014.

[24] G. Huang, M. Ramesh, T. Berg and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *University of Massachusetts, Amherst,* 2007.

[25] G. Huang, M. Mattar, H. Lee and E. Learned-Miller, "Learning to align from scratch," *NIPS,* 2012.

[26] K. Simonyan, A. Vedaldi and A. Zisser, "Learning local feature descriptors using convex optimisation," *Pattern Analysis and Machine Intelligence,* pp. 1573-1585, 2014.

[27] M. Grgic, K. Delac and S. Grgic, "SCface–surveillance cameras face database," *Multimedia tools and applications ,* vol. 51, no. 3, pp. 863-879, 2011.

[28] R. Wallace and M. McLaren, "Inter-session variability modelling and joint factor analysis for face authentication," *Biometrics (IJCB), International Joint Conference on,* 2011.

[29] J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, "STRIVING FOR SIMPLICITY: THE ALL CONVOLUTIONAL NET," *ICLR ,* 2015.

[30] B. Graham, "Fractional Max-Pooling," *arXiv:1412.6071 [cs.CV],* 2015 .

[31] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun,* pp. 17: 1-2, 1995.

[32] P. Mermelstein, "Distance Measures for Speech Recognition--Psychological and Instrumental," *Joint Workshop on Pattern Recognition and Artificial Intelligence,* 1976.

[33] R. D. P. K. N. B. P. O. P. D. Najim Dehak, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification," *Interspeech,* 2009.

[34] D. a. M. A. a. K. J. Wu, "MLP Internal Representation as Discriminative Features for Improved Speaker Recognition," *Proceedings of the 3rd International Conference on Non-Linear Analyses and Algorithms for Speech Processing,* no. NOLISP'05, pp. 72--80, 2005.

[35] A. C. M. a. J. K. Dalei Wu, "MLP Internal Representation as Discriminative Features for Improved Speaker Recognition".

[36] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study.," *Proceedings of the 26th European Conference on Information Retrieval (ECIR),* 2004.

[37] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents.," 2013.

[38] A. L. Mass, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, "Learning Word Vectors for Sentiment Analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* pp. 142--150, 2011.

[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, " Distributed representations of phrases and their compositionality," *In Advances on Neural Information Processing Systems,* 2013.

[40] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive Science,* pp. 34: 1388-1429, 2010.

[41] F. Zanzotto, I. Korkontzelos, F. Fallucchi and S. Manandhar, "Estimating linear," *COLING,* 2010.

[42] A. Yessenalina and C. Cardie, "Compositional matrix-space models for sentiment analysis," in *Empirical Methods in Natural Language Processing*, 2011.

[43] H. Larochelle and S. Lauly, "A neural autoregressive topic model," in *Advances in Neural Information*, 2012.

[44] E. Grefenstette, G. Dinu, Y. Zhang, M. Sadrzadeh and M. Baroni, "Multi-step regression learning for compositional distributional semantics," in *Empirical Methods in Natural Language Processing*, 2013.

[45] Y. Bengio, H. Schwenk, S. Senecal, F. Morin and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*, Springer, 2006, p. 137–186.

[46] T. Mikolov, "Statistical Language Models based on Neural Networks," PhD thesis, Brno University of Technology, 2012.

[47] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and text classification," in *Proceedings of the 50th Annual Meeting of the Association*, 2012.

[48] G. E. Dahl, R. P. Adams and H. Larochelle, "Training Restricted Boltzmann Machines on word observations," in *International Conference on Machine Learning*, 2012.

[49] R. Socher, D. Chen, C. D. Manning and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion.," in *Advances in Neural Information Processing Systems*, 2013.

[50] R. Socher, E. H. Huang, J. Pennington, C. D. Manning and A. Y. Ng, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in Neural Information Processing Systems*, 2011.

[51] Y. Ko, "A study of term weighting schemes using class information for text classification," *SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval,* pp. 1029-1030, 2012.

[52] Z. Harris, "Distributional Structure," *Word,* p. 10 (2/3): 146–62, 1954.

[53] D. D. Lewis, Y. Yang, T. G. Rose and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research,* pp. 5: 361-397, 2004.

[54] J. Leskovec, A. Rajaraman and J. D. Ullman , Mining of Massive Datasets, 2nd ed., Cambridge University Press, 2014.

[55] T. Mikolov, K. Chen , G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[56] A. Graves , A.-r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," *ICASSP IEEE International Conference,* pp. 6645-6649, 2013.

[57] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "Web-scale training for face identification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2015.

[58] Y. Sun, Y. Chen, X. Wang and X. Tang, "Deep learning face representation by joint identification-verification," *Advances in Neural Information Processing Systems,* 2014.

[59] Y. Sun, X. Wang and X. Tang, "Deep learning face representation from predicting 10,000 classes," *CVPR,* pp. 1891-1898, 2014.

[60] Y. Sun, X. Wang and X. Tang, "Deeply Learned Face Representations Are Sparse, Selective, and Robust," *CVPR,* pp. 2892-2900, 2015.

[61] Y. Sun, D. Liang, X. Wang and X. Tang, "DeepID3: Face Recognition with Very Deep Neural Networks," *CVPR,* 2015.

[62] F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CVPR,* 2015.

[63] G. B. Huang, M. Ramesh, T. Berg and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst 07-49, 2007.

[64] D. Chen, X. Cao, L. Wang, F. Wen and J. Sun, "Bayesian face revisited: A joint formulation," in *Computer Vision – ECCV 2012*, Springer, 2012, p. 566–579.

[65] Y. Sun, X. Wang and X. Tang, "Deep learning face representation by joint identification - verification," *arXiv,* no. 1406.4773, 2014.

[66] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[67] "CARC: http://bcsiriuschen.github.io/CARC/," [Online]. Available: http://bcsiriuschen.github.io/CARC/.

[68] Y. Dong, L. Zhen, L. Shengcai and . Z. L. Stan, "Learning Face Representation from Scratch," *arXiv preprint ,* no. 1411.7923, 2014.

[69] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009.

[70] G. Gregory, A. Holub and P. Perona, "Caltech-256 object category dataset," 2007.

[71] A. Coates, Y. N. Andrew and L. Honglak, "An analysis of single-layer networks in unsupervised feature learning," in *International conference on artificial intelligence and statistics*, 2011.

[72] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV),* vol. 115, no. 3, pp. 211-252, 2015.