ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2023.0416

A DEEP LEARNING BASED ALGORITHM FOR IMPROVING EFFICIENCY IN MULTIMEDIA APPLICATIONS

R. Jayadurga[1], M. Sathiya[2] and G.K. Arpana[3]
[1]*Department of Computer Science, Soundarya Institute of Management and Science, India*
[2]*Department of Information Technology, Karpagam Institute of Technology, India*
[3]*Department of Electronics and Communication Engineering, East West College of Engineering, India*

Abstract

Most of the time, these classifiers are trained using general-purpose datasets with a lot of classes. Therefore, the performance of these classifiers may not be as good as it could be. Both choosing classifiers based on registrations and dividing them into groups based on the subjects they cover are possible solutions that could lead to better classifier performance. This makes it clear that a classifier division and selection strategy needs for the proposed optimization to work. With the help of this method, the proposed model for feature extraction can choose an appropriate classifier while taking subscription constraints into account. There are subscriptions with the best values of n, and the results of using only n-class classifiers from one domain and ignoring classes from other domains are also given. These are in the same place as the effects of only using n-class classifiers from a certain domain. In this article, these are talked about in the same context as what happens when you only use n-class classifiers from a certain domain. For high-performance use of SAE-based systems, you need to use a classifier selection technique. This method is also needed for the investigation of multimedia events that need the method. To establish the effectiveness of the multimedia event-based system as well as its dependability, we are making use of traditional evaluation methods such as throughput and accuracy. These measures include the following: When compared to the efficiency of the system when using a classifier with a single class, the efficiency of the system diminishes as the number of classes per classifier increases. This is the case regardless of the other measures. This is the situation about both the throughput and the precision of the operation.

Keywords:

Multimedia Data, Stacked Auto Encoder, Deep Learning, Classifier

1. INTRODUCTION

The Internet of Thing (IoT) has been developed so that it can support smart devices. To bridge the functionality gap that occurs between the software that makes the IoT usable and the IoT themselves, event-based systems are currently under development. Event-driven analytics are reliant on accumulation and examination (processing) of structured data streams [1].

This is the fundamental principle upon which event-driven statistics are built. The publish/subscribe paradigm is the foundation of event processing systems because it makes it possible to streamline communication between individuals who produce content and individuals who make use of the work that the individuals who produce content create. The purpose of developing event processing systems is to process a user subscription in the shape of a language for rules and investigate the structured events [2].

In addition, the different kinds of smart city devices are responsible for the production of significant quantities of unstructured data in the form of multimedia. One example of a structured event is the processing of readings received from sensors such as those that measure temperature or energy. The events that are generated by traffic cameras, on the other hand, are examples of unstructured events that are associated with traffic consciousness.

The event-based systems that are in use today do not support the processing of such events, there is a need for an event processing system that is based on the Internet of Multimedia Things (IoMT) and is also capable of processing images and videos. This is because the processing of such events is not supported by the event-based systems that are in use today. The IoMT is an IoT-based paradigm that enables objects to connect with each other and share structured and unstructured data.

To better enable multimedia-based services and applications in smart cities [3]. IoMT can be described as an IoT-based paradigm that enables objects to connect with each other and share structured and unstructured data. This is done to improve the accessibility of services and applications dependent on multimedia content.

Object detection in images is a frequent problem in the field of image processing [4], which is why smart cities are an appropriate setting for this research. Real-time image-based systems are currently available in a vast majority, and most of these real-time image-based systems are exceptionally excellent at recognising objects. This is due to the features that are specific to their respective domains. It has been demonstrated that deep convolutional neural networks [5] are well adapted for the task of image classification, with remarkable outcomes.

There are currently no multimedia query languages that are available for events, and the event processing systems that are currently in use [6] are not designed to deal with the unstructured event types that are produced by the IoMT. In addition, there are currently no multimedia query languages that are accessible for events. Even though most image processing systems [7] [8] are built without taking the event-based paradigm into consideration during the construction process, these systems are still very good at recognising objects in image events and are highly specialised for the fields in which they are used.

The user expressiveness is seriously constrained because image processing systems do not offer much in the way of a query language. In today world, programmers are required to build an entirely new application each time they want to combine the outcomes of two separate processing systems, such as event processing and image processing. It is essential that they have this capability for them to be able to efficiently consume info pertaining to multimedia events. The expenses associated with the initial setup are high, and there are difficulties associated with combining the results of the two different systems. It is essential to have an IoMT-aware event processing engine to facilitate

adaptive multimedia services among the applications that are used in smart cities and the various event environments that they operate in.

To achieve the goal of processing and analysing IoMT-based events as native events while maintaining a high level of performance, the event-based multimedia stream processing engine (MSPE) is proposed. This is done in conjunction with an optimization technique that is comprised of neural network-based feature extraction operators. This adds support for the examination of multimedia content within event-based systems to the event query language. The user of MSPE can create subscriptions within the software with the assistance of the recommended detect operator, which is founded on object recognition.

The subscription will be utilised by the object identification model to determine which classifier is the best one to use when searching for the designated attribute. We came up with an optimization model that makes use of subscriptions in two distinct ways: first, to analyse the query, and second, to optimise the processing of a neural network-based matcher according to the constraints imposed by subscriptions. This model was developed in response to a challenge posed by a client who wanted to improve the performance of their matching system.

It can effectively manage multimedia event streams coming from a variety of apps while maintaining high throughput and comparable accuracy, the model that was developed as a result is ideal for implementation in the infrastructure of smart cities. This is because it is ideal for implementation in the infrastructure of smart cities because it is ideal for implementation in the infrastructure of smart cities.

2. RELATED WORKS

Research on training algorithms, which was the focus of methods that took an algorithmic or model-centric strategy, had as their primary goal improved performance in the classification of imbalanced data. This was the primary objective of the research into and iteration on training algorithms. This objective was accomplished through the processes of conducting research on training algorithms and iterating on those algorithms. For instance, cost-sensitive learning techniques aim to maximise loss functions associated with a data collection to enhance classification performance. This is done to reduce the overall cost of the learning process [5].

The realisation that the costs of incorrect classifications vary significantly across different types of applications in the real world served as the inspiration for the development of these learning strategies. The goal of these strategies is to help students improve their classification skills. The first thing that needs to be done to establish the cost matrix when these techniques are applied to the learning phase is to look at the data [6].

The concept of cost-sensitive learners is closely related to the strategy of shifting the power dynamic in favour of the individual. There has been some research that has shown potential in improving classification performance on imbalanced data; however, this research is neither comprehensive nor methodical. Some studies have indicated that there is room for improvement in classification effectiveness with imbalanced data [7].

The term data transformation has the potential to be utilised in a wide variety of contexts and context-specific contexts. Oversampling and undersampling are two sampling-based methods that have garnered a lot of attention in the battle against the effects of skewed data. Both methods are designed to combat the effects of skewed data. Both approaches require collecting either an excessively large or an insufficiently large number of samples [8].

The topic of oversampling versus downsampling for unbalanced data sets has been the subject of a great number of studies, and these studies have generated a wide variety of findings and (sometimes) conflicting opinions. Oversampling is a technique that is used to achieve statistical parity by having numerous identical or nearly identical positive data instances generated by a set of algorithms. This technique is called oversampling, and it is one of the techniques that can be used to achieve statistical parity. One of the methods that sees the most action in the industry is called sampling with replacement, and it goes by that name for a reason. An improved method of oversampling that is based on synthetic minority oversampling was introduced [9].

On the other hand, there is a probability of overfitting when there are an excessive number of samples collected. This happens when there are too many samples. Downsampling involves selecting a subgroup of negative samples (data examples) to construct a model utilising the same number of positive samples. This is done to maintain the integrity of the data. The fact that it only employs a small proportion of the total population is one factor that contributes to the high degree of efficiency it possesses. The most significant drawback is that there is a probability that information will be lost because many data occurrences in the majority class will be ignored. This is the result of many data occurrences being ignored [10].

Easy Ensemble begins by taking several random samples from the larger group, then trains a separate classification algorithm on each of those samples, and ultimately combines the predictions that were generated by each of those algorithms. In Balance Cascade, the models are put through their trials in a specific order to evaluate their performance. Data instances that have been correctly classified by the models that are currently being trained will not be included in subsequent rounds if they pertain to the majority class and have been correctly classified [11].

Deep learning, whose fundamental concept originated from research on artificial neural networks, has drawn significant research efforts in a wide variety of fields within the past few years due to its ability to achieve top performance for a variety of tasks. This is since deep learning can learn how to achieve optimal results for a wide range of tasks. This is because deep learning has the capability of achieving top performance for a broad range of activities. This article presents a more in-depth investigation into the research that is currently being done on deep learning.

In particular, the Convolutional Neural Network (CNN), which is categorised as a discriminative deep architecture within the DNN category, has demonstrated outstanding performance in a variety of computer vision and image recognition competitions. These competitions include the ImageNet Challenge, the ImageNet Cup, and the ImageNet Challenge. Pooling layers and convolutional layers are the two types of modules that are utilised

in the construction of convolutional neural networks, which are also referred to as CNNs.

It is standard procedure to layer each of these components one on top of the other when attempting to create a model with a high level of precision. To reduce the overall input rate, the pooling layer is responsible for accepting a smaller subsample of the output from the convolutional layer. The weights for the convolutional layer are distributed among the layers that come after it in the neural network.

Even though CNNs have shown promising results for classification tasks in many applications, it is still unknown how well they will perform on data sets that are extremely imbalanced. This is the case even though it is known that CNNs will perform well on data sets that are extremely unbalanced. In this paper, we investigate the applicability of CNNs for the classification of asymmetric data and, more significantly, how to improve their overall performance by expanding the scope of their capabilities.

One of the expansions that we have proposed for CNNs is the incorporation of the method with a bootstrapping sampling strategy that is specifically designed to satisfy the requirements of CNNs. This is one of the expansions that we have proposed. Our proposed bootstrapping sampling approach combines oversampling with decision fusion to improve CNN performance on the classification of multimedia data with or without imbalanced data distributions. This is in contrast to the negative bootstrapping method, which combines random sampling with adaptive selection to iteratively find relevant negatives. This helps improve CNN ability to categorise data regardless of the data distribution, which helps improve CNN ability to use the data.

3. DEEP AUTO ENCODER

Auto Encoder (AE) needs to be laid out for us before we can fully understand what a stacked autoencoder (SAE). AE are a type of deep learning architecture that are trained to encode and decode data by utilising unsupervised training techniques. Autoencoders are also used for the purpose of encoding data. They are very comparable to ANNs, which are a different kind of architecture for machine learning. An AE network is depicted in Fig.1.

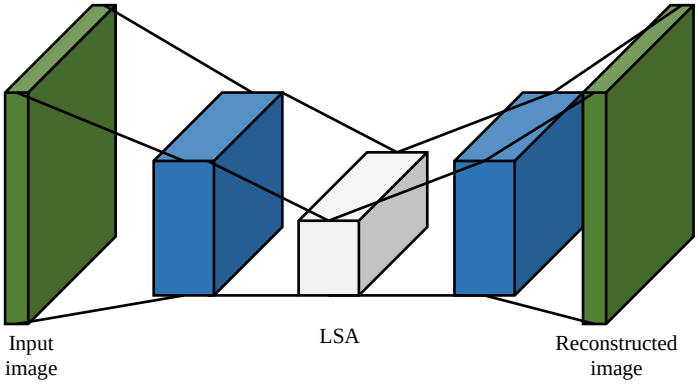

Fig.1. Autoencoder Network

An AE network, just like an ANN is made up of an input layer, a concealed layer, and an output layer. The Fig.1 shows an AE network. Each layer of an autoencoder contains a particular quantity of synapses that has been established in advance. CNN outperform feed-forward neural networks, which have a greater number of neurons in their input and output layers but a lesser number of neurons in their hidden layer.

The performance of convolutional neural networks is superior to that of feed-forward neural networks. There are a wide variety of autoencoders, but the following are some of the more popular ones: sparse AE, zero-bias AE, denoising AE, contractive autoencoder, and convolutional AE.

There are three levels in an autoencoder network: the input layer, the hidden layer, and the output layer. The operation of the AE system is depicted in a straightforward manner in Fig.1, which can be found here. An input, let say x, is given to the autoencoder as the procedure starts off to get things started. This input vector is then accepted by the encoder component, which employs it in the process of constructing the secret code after having received it. The decoder mechanism receives this top-secret code to use as input in its operation.

The decoder job is to take the code and produce an output that can be used to reconstitute the original input, which was x'. The autoencoder primary function is to filter out noise in the data and extract relevant features from large datasets. During the encoding phase, the Eq.(1) is used to acquire the label, and during the decoding phase, the Eq.(2) is used to reconstitute the original data, which is x'. After the error has been calculated utilising Eq.(3), the backpropagation algorithm can use it to fine-tune the network and bring the reconstructed output as close as it can get to the input data. This is accomplished by bringing the reconstructed output as close as it can get. The methodology that was just explained has as its primary goal the identification of significant patterns within the data that was provided [26].

$$x'=F(wc+b') \quad (1)$$

$$c=F(wtx+b) \quad (2)$$

$$e=\min n(x'-x)^2 \quad (3)$$

The character represents the activation functions (F) that are used in Eq.(1) and Eq.(2). These equations can be found below. The value b stands for the bias, and w and z stand for the weights that are applied to the input layer and the concealed layer to make the connection between the two. The outcome of having c reconstructed is the value that is indicated by the parameter x', which is the result that is achieved after having c reconstructed. Overfitting, which is demonstrated by the equation, is also present in ANN; however, it can be avoided through the application of regularisation (4).

$$\min[\{\sum i=1n(x'-x)2\}+\gamma L(w)] \quad (4)$$

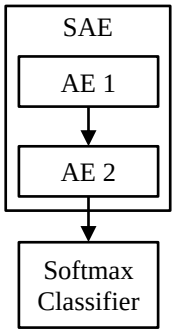

Fig.3. SAE

The parameter for weight correction is expressed as $L(w)$ and the γ parameter for regularisation is represented by the parameter.

The values of these constraints are established through the utilisation of a procedure that is commonly referred to as the hit-and-miss approach.

An illustration of an example of a deep layered autoencoder model is provided in Fig.2. The weights of a deep neural network are originally initialised using this model by stacking several autoencoders in a greedy fashion, building one layer on top of the one that came before it.

The Fig.3 depicts a stacked autoencoder construction for its reader perusal. This architecture is made up of two autoencoders that are organised in a cascade fashion. Its purpose is to decrease the dimensions of the data by selecting only the features that are relevant to the issue at hand.

These features can then be entered into a softmax classifier to address any classification issues that may have been encountered. After the data that has been received has been encoded by the stacked autoencoder, the softmax layer is utilised to categorise the data that has been obtained.

It is a linear classification that accomplishes its goals by calculating the probability distribution of the input across n different sources. It achieves this goal by placing a greater priority on relevant characteristics, which ultimately results in an improvement in classification accuracy.

$$F(x_i)=Exp(x_i)\sum exp(x_j) \quad (5)$$

The softmax function generates predictions that are based on the exponential of the value x_i that is passed in, the sum of the exponentials of all the values that are passed in, and the percentage of these. The softmax function will return the probabilities that are affiliated with each class if there is an issue with multiple classifications. It is anticipated that the output class will have the highest probability in this situation.

4. PROPOSED MODEL

Before running the data through a technique that will classify it, it is best practise to first clean and modify the data. You will need to examine any imbalanced data, estimate any missing values, remove noisy data such as outliers, and normalise the data as part of the pre-processing portion of the project.

It is impossible to prevent the occurrence of missing numbers when dealing with information that is based on the real world. To handle them in the simplest way possible, you can simply disregard the entirety of the record if it has any missing values. This will allow you to deal with them as efficiently as possible.

This should not be used with libraries that have fewer records. We don't have to get rid of records that are missing data because we can use data imputation algorithms to fill in the blanks of records that are lacking data. This keeps us from having to throw away records that are missing data. We can replace missing numerical characteristics with the number that is generally associated with that attribute if we make use of a technique called median imputation. In the case of nominal attributes, on the other hand, we can fill in the blanks with the attribute most frequent value by making use of a technique that is referred to as mode interpolation.

After the preparatory steps had been completed, the dataset was then segmented into a collection that would be used for training and another that would be used for testing. The proposed model for the classification of CKD is presented in Fig.3, which illustrates the comprehensive process flow of the model.

The layered autoencoder was fed the information that was collected from the dataset pertaining to multimedia dataset. Following the completion of the training for the input characteristics, the data are afterwards inserted into the AE1 system. Both the extraction of features and the reduction of the dimensionality of the data will be finished simultaneously because of this procedure.

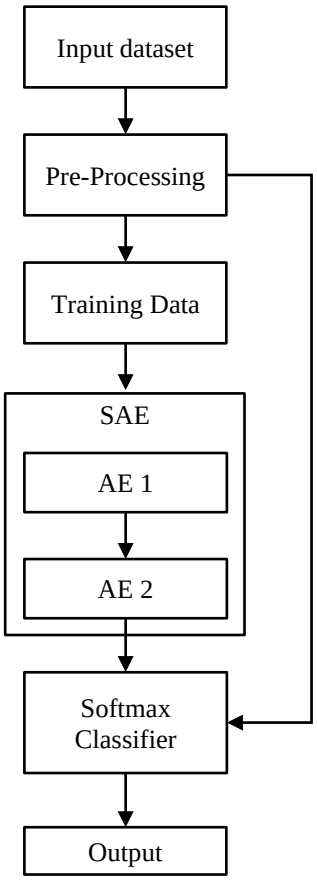

Fig.3. Proposed Framework

After the possibilities associated with each class have been calculated, the results of AE2 are used as input into a SoftMax classifier. This follows the step in which the results of AE1 were used. The classification and output associated with the class that has the highest probability of being accurate are then provided by this classifier.

5. RESULTS AND DISCUSSION

The results of 180 videos of testing that are listed in Table.1-Table.6. The features are fed into the AE1 (autoencoder 1) To produce a synopsis of the 10 characteristics that are thought to be the most important. AE2, which is the second autoencoder, takes this information as input and uses it to generate the model ten hidden neurons.

The information that is provided by AE1 is used to generate the model hidden neurons. In addition, the corresponding parameters are generated with the help of this information. Following that, the results of AE2 were incorporated into an

algorithm known as Softmax. Overfitting was identified as a potential issue, and regularisation and 5-fold confirmation were beneficial in preventing its occurrence.

The parameter values for L2WeightRegularization and Sparsity Regularization were both adjusted to have a value of 0.01, correspondingly. In terms of the sparsity percentage, it was determined that a cutoff of 0.05 would serve as the benchmark. The procedures were given a time limit of 2 milliseconds, and the number of repetitions was increased to 100 so that an accurate prediction of the results could be made.

The usefulness of a deep stacked autoencoder model can be decided based on its accuracy, specificity, precision, and recall values, in addition to its F1-score. The TRECVID data set is a large benchmark data set that is distinguished by an extremely asymmetrical distribution of training and test data. This is the TRECVID data set primary differentiating feature. This data collection is used to evaluate the efficacy of our proposed framework for the classification of multimedia data.

Using the IACC.1.B dataset that was made available by the TRECVID 2011 benchmark [38], we conduct an analysis of SAE for the purpose of this investigation. A technology known as SIN has the potential to be of great assistance in the process of video retrieval, classification, and other application areas. The algorithm deciphers the meaning of the information that it contains. Some instances of high-level semantic concepts include vehicle, road, and tree. Some of the difficulties that it presents include the unpredictability of the data, an incapacity to scale it, and a disconnect between the meaning and the language.

Table.1. Accuracy of Improvement

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	91.68	90.74	82.01	89.43	87.39	92.93
20	92.51	95.73	88.32	93.45	98.84	94.88
30	90.92	92.22	88.40	92.72	82.92	92.99
40	89.19	82.47	91.07	92.35	86.71	90.17
50	88.11	87.32	98.11	94.91	89.55	90.62
60	88.83	87.45	86.72	99.66	97.61	90.64
70	92.01	72.65	99.96	85.05	99.50	91.17
80	90.12	93.43	93.89	85.05	93.34	92.91
90	91.16	94.52	93.17	73.74	92.87	93.76
100	91.68	90.74	82.01	89.43	87.39	92.93
110	89.62	88.70	80.17	87.42	85.43	90.84
120	90.43	93.58	86.33	91.35	96.62	92.75
130	88.88	90.15	86.41	90.64	81.06	90.90
140	87.18	80.62	89.02	90.27	84.76	88.14
150	86.13	85.36	95.90	92.78	87.54	88.58
160	86.83	85.48	84.77	97.42	95.42	88.60
170	89.94	71.02	97.71	83.14	97.26	89.12
180	88.09	91.33	91.78	83.14	91.24	90.82

Table.2. Sensitivity of Improvement

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	89.011	88.097	79.621	86.825	84.845	90.223
20	89.815	92.942	85.748	90.728	95.961	92.117
30	88.276	89.534	85.825	90.019	80.505	90.282
40	86.588	80.068	88.417	89.660	84.184	87.544
50	85.545	84.777	95.252	92.146	86.942	87.981
60	86.240	84.903	84.194	96.757	94.767	88.000
70	89.329	70.534	97.049	82.573	96.602	88.515
80	87.491	90.709	91.155	82.573	90.621	90.204
90	88.504	91.767	90.456	71.592	90.165	91.029
100	89.011	88.097	79.621	86.825	84.845	90.223
110	87.010	86.116	77.831	84.873	82.937	88.195
120	87.796	90.852	83.820	88.688	93.804	90.045
130	86.291	87.521	83.896	87.996	78.695	88.252
140	84.641	78.268	86.430	87.644	82.292	85.575
150	83.621	82.871	93.111	90.074	84.987	86.003
160	84.301	82.994	82.301	94.582	92.636	86.022
170	87.320	68.948	94.867	80.716	94.430	86.524
180	85.524	88.669	89.106	80.716	88.584	88.176

Table.3. Specificity of Improvement

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	87.725	86.824	78.471	85.571	83.619	88.920
20	88.518	91.599	84.509	89.417	94.575	90.786
30	87.001	88.240	84.585	88.719	79.342	88.977
40	85.336	78.911	87.140	88.365	82.968	86.279
50	84.309	83.552	93.876	90.814	85.686	86.709
60	84.994	83.676	82.978	95.359	93.398	86.729
70	88.038	69.515	95.646	81.380	95.206	87.236
80	86.227	89.398	89.838	81.380	89.312	88.901
90	87.226	90.441	89.149	70.558	88.862	89.714
100	87.725	86.824	78.471	85.571	83.619	88.920
110	85.753	84.872	76.707	83.647	81.739	86.921
120	86.528	89.539	82.609	87.407	92.448	88.744
130	85.044	86.256	82.683	86.724	77.558	86.977
140	83.418	77.137	85.181	86.378	81.103	84.339
150	82.413	81.673	91.766	88.773	83.759	84.760
160	83.083	81.795	81.112	93.215	91.298	84.779
170	86.059	67.952	93.496	79.550	93.066	85.274
180	84.289	87.388	87.818	79.550	87.304	86.902

Table.4. F-Measure of Improvement

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	89.389	88.472	79.960	87.194	85.205	90.607
20	90.197	93.337	86.112	91.114	96.369	92.508
30	88.651	89.915	86.190	90.402	80.847	90.665
40	86.956	80.408	88.793	90.041	84.542	87.916
50	85.908	85.137	95.657	92.537	87.311	88.355
60	86.606	85.264	84.552	97.169	95.170	88.374
70	89.708	70.834	97.461	82.924	97.013	88.891
80	87.863	91.094	91.543	82.924	91.007	90.587
90	88.881	92.157	90.841	71.897	90.548	91.416
100	89.389	88.472	79.960	87.194	85.205	90.607
110	87.380	86.482	78.162	85.234	83.290	88.570
120	88.169	91.238	84.176	89.065	94.202	90.428
130	86.658	87.893	84.252	88.370	79.029	88.627
140	85.001	78.601	86.797	88.017	82.642	85.939
150	83.977	83.223	93.507	90.457	85.348	86.368
160	84.659	83.347	82.651	94.984	93.030	86.387
170	87.692	69.241	95.270	81.059	94.831	86.892
180	85.888	89.046	89.485	81.059	88.960	88.551

Table.5. MAPE

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	12.301	13.202	21.552	14.455	16.406	11.107
20	11.509	8.428	15.516	10.609	5.453	9.241
30	13.025	11.786	15.440	11.308	20.682	11.049
40	14.689	21.112	12.886	11.661	17.056	13.747
50	15.716	16.473	6.152	9.213	14.340	13.316
60	15.031	16.349	17.047	4.669	6.630	13.297
70	11.988	30.506	4.382	18.644	4.822	12.790
80	13.798	10.628	10.188	18.644	10.714	11.126
90	12.800	9.586	10.877	29.463	11.164	10.313
100	12.301	13.202	21.552	14.455	16.406	11.107
110	14.273	15.153	23.316	16.378	18.285	13.105
120	13.498	10.487	17.416	12.619	7.579	11.282
130	14.981	13.769	17.341	13.302	22.465	13.049
140	16.607	22.886	14.844	13.648	18.921	15.686
150	17.611	18.351	8.262	11.254	16.266	15.265
160	16.942	18.229	18.912	6.812	8.729	15.247
170	13.967	32.068	6.532	20.473	6.962	14.751
180	15.736	12.638	12.208	20.473	12.722	13.124

Table.6. MAE

Video	ANN	CNN	AlexNet	VGG	SAE1	SAE2
10	13.016	13.909	22.192	15.152	17.087	11.831
20	12.230	9.175	16.205	11.338	6.224	9.981
30	13.734	12.505	16.129	12.030	21.328	11.774
40	15.384	21.755	13.596	12.381	17.732	14.450
50	16.403	17.154	6.917	9.953	15.038	14.023
60	15.724	17.030	17.723	5.446	7.391	14.004
70	12.705	31.072	5.161	19.307	5.598	13.501
80	14.501	11.357	10.920	19.307	11.442	11.850
90	13.511	10.323	11.603	30.038	11.888	11.044
100	13.016	13.909	22.192	15.152	17.087	11.831
110	14.972	15.844	23.941	17.059	18.951	13.813
120	14.203	11.217	18.089	13.331	8.332	12.005
130	15.674	14.472	18.015	14.008	23.097	13.758
140	17.287	23.514	15.538	14.351	19.582	16.373
150	18.283	19.016	9.009	11.977	16.948	15.956
160	17.619	18.896	19.573	7.572	9.473	15.937
170	14.668	32.622	7.294	21.122	7.720	15.446
180	16.423	13.350	12.923	21.122	13.433	13.832

An illustration of three keyframes, along with their respective explanations, can be found here. Conventional deep learning techniques, such as CNNs, have difficulty functioning well on the TRECVID data set for several reasons. These reasons include under-fitting, huge diversity, noisy and incomplete data annotation, and other similar issues. The TRECVID data collection contains ideas with various degrees of data imbalance, it is possible that a single batch size would not be optimal for all of them. This raises the possibility that a different batch size would be optimal for each of the ideas.

The fact that the collection includes ideas with varying degrees of data imbalance makes the occurrence of this possibility a distinct possibility. Because of this, the size of the batches is determined in real time based on the proportion of successful training examples to the total number of examples in the collection. During our experiment, we made use of a batch size that was twice as large as the number of training instances that were considered as being successful. This allowed us to replicate the results of our experiment more accurately.

Each occurrence is used not once, but twice: once for instruction (100K), and then again for evaluation (50K). Recall is given more weight than accuracy in terms of classification metrics when working with skewed data sets; the F-score represents this trade-off because it is dependent on recall.

The data annotations are both noisy and incomplete, our F-scores are higher for both frameworks. Neither of them can identify any genuine positive instance, which is why F-scores are higher. The fact that CNNs performed so poorly in comparison to the other classifiers that were evaluated on the TRECVID dataset is additional evidence that the combination of SAEs and the bootstrapping technique in our framework for the classification of

imbalanced multimedia data is extremely effective. This evidence comes from the fact that CNNs performed so poorly on the TRECVID dataset.

6. CONCLUSION

In this article, SAE is developed by combining them with an approach known as bootstrapping. The process of bootstrapping entails generating several pseudo-balanced training samples based on the characteristics of the data collection. These quantities are put through the SAE classification process so that they can be sorted. Our proposed approach is successful in classifying multimedia data even though it has a highly asymmetric distribution, as shown by experiments that were carried out on the TRECVID data set. In contrast to many other studies, which use raw media data as the input, our extended SAE framework has been demonstrated to function successfully on low-level features. The total quantity of training time that must be completed to achieve deep learning is significantly cut down.

REFERENCES

[1] V. Saravanan and M. Rizvana, "Dual Mode Mpeg Steganography Scheme for Mobile and Fixed Devices", *International Journal of Engineering Research and Development*, Vol. 6, pp. 23-27, 2013.

[2] V. Saravanan and C. Chandrasekar, "Qos-Continuous Live Media Streaming in Mobile Environment using VBR and Edge Network", *International Journal of Computer Applications*, Vol. 53, No. 6, pp. 1-12, 2012.

[3] A.N. Reddy and J.C. Wyllie, "I/O Issues in a Multimedia System", *Computer*, Vol. 27, No. 3, pp. 69-74, 1994.

[4] H. Babbar and S. Rani, "A Genetic Load Balancing Algorithm to Improve the QoS Metrics for Software Defined Networking for Multimedia Applications", *Multimedia Tools and Applications*, Vol. 81, No. 17, pp. 9111-9129, 2022.

[5] M.K. Gupta and P. Chandra, "Effects of Similarity/Distance Metrics on K-Means Algorithm with Respect to its Applications in IoT and Multimedia: A Review", *Multimedia Tools and Applications*, Vol. 81, No. 26, pp. 37007-37032, 2022.

[6] X. Zhang, "Intelligent Recommendation Algorithm of Multimedia English Distance Education Resources based on User Model", *Journal of Mathematics*, Vol. 2022, pp. 1-8, 2022.

[7] Z. Lv and A. Alamri, "Deep Learning-based Smart Predictive Evaluation for Interactive Multimedia-Enabled Smart Healthcare", *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 18, No. 1, pp. 1-20, 2022.

[8] A.A. Khan and S. Karim, "IPM-Model: AI and Metaheuristic-Enabled Face Recognition using Image Partial Matching for Multimedia Forensics Investigation with Genetic Algorithm", *Multimedia Tools and Applications*, Vol. 81, No. 17, pp. 23533-23549, 2022.

[9] C. Peng, "An Application of English Reading Mobile Teaching Model based on K-Means Algorithm", *Mobile Information Systems*, Vol. 2022, pp. 1-14, 2022.

[10] A. Hafsa, "Real-Time Video Security System using Chaos-Improved Advanced Encryption Standard (IAES)", *Multimedia Tools and Applications*, Vol. 56, pp. 1-24, 2022.

[11] M.A.R. Khan, V.J. Tharini and M.B. Alazzam, "Optimizing Hybrid Metaheuristic Algorithm with Cluster Head to Improve Performance Metrics on the IoT", *Theoretical Computer Science*, Vol. 927, pp. 87-97, 2022.