EasyChair Preprint

№ 6118

Classifying Textual Data with Pre-Trained Vision Models Through Transfer Learning and Data Transformations

Charaf Eddine Benarab

July 23, 2021

Classifying Textual Data with Pre-trained Vision Models through Transfer Learning and Data Transformations

Charaf Eddine Benarab
University of Electronics Science and Technology of China
School of Computer Science and Engineering
Chengdu, China
charafeddineben@std.uestc.edu.cn

***Abstract*—Knowledge is acquired by humans through experience, and no boundary is set between the kinds of knowledge or skill levels we can achieve on different tasks at the same time. When it comes to Neural Networks, that is not the case, the breakthroughs in the field are extremely task and domain-specific. Vision and language are dealt with in separate manners, using separate methods and different datasets. Current text classification methods, mostly rely on obtaining contextual embeddings for input text samples, then training a classifier on the embedded dataset. Transfer learning in Language related tasks, in general, is heavily used in obtaining the contextual text embeddings for the input samples. In this work, we propose to use the knowledge acquired by benchmark Vision Models which are trained on ImageNet to help a much smaller architecture learn to classify text. A data transformation technique is used to create a new image dataset, where each image represents a sentence embedding from the last six layers of BERT projected on a 2D plane using a t-SNE based method. We trained five models containing layers sliced from vision models pre-trained on ImageNet on the created image dataset for the IMDB dataset embedded with the last six layers of BERT. Despite the challenges posed by the very different datasets, experimental results achieved by this approach which links large pre-trained models on both language and vision, are very promising, without needing high compute resources. Specifically, Sentiment Analysis is achieved by five different models on the same image dataset obtained after BERT embeddings are transformed into gray scale images.**

***Index Terms*—Natural language processing, Text Classification, Image Classification, t-SNE, BERT, Transfer Learning, Convolutional Neural Networks, Domain Adaptation**

I. Introduction

Attention-based architectures and specifically the Transformer [25] sparked a revolution in the world of Natural Language Processing and Deep Learning in General. BERT-representations [5] opened a lot of new goals and challenges for the Machine Learning Community, because of their highly semantic embeddings and the kind of knowledge they encode given textual data. Being pre-trained to predicted Masked Language and next sentences, and on a huge Wikipedia Corpus, makes it a powerful benchmark for Natural Language Processing and a current standard for word and sentence representations. Computer Vision made very remarkable achievements [29]. The seminal work introduced in AlexNet [14] and the parallel structure of Convolutional Neural Networks enabled the use of GPUs, then it became a standard for training Neural Networks for Visual Recognition. Different architectures have emerged [14] [15] [21] [9] [27] since then using CNN's as a building block and achieving higher accuracies on different datasets. Transfer Learning [1] [2] opened the doors for a very wide range of applications, allowing for the exchange of previously acquired knowledge from a large dataset on a certain task to another task with a much smaller dataset. This philosophy is a current essential practice in both academia and industry as it helps avoid a very long and computationally demanding procedure. In Visual Understanding tasks, transfer learning in the last decade heavily relied on ImageNet [4] as a base or $Source$ dataset, ImageNet [4] is a huge dataset containing over 14 million images and their different one-thousand class labels and is considered the de facto for pre-training Vision Models. The work described in this paper aims to use knowledge acquired by vision models to train text-classifiers for Sentiment Analysis, using a t-SNE based transformation method brought about in [19] to transform IMDB Text Embeddings from BERT into Gray Scale Images. The main objective of this paper is to bring $Language$ and $Vision$ a step closer, and to harness the power of transfer learning from large image datasets in Natural Language Processing and vice-versa. Hoping this would be an opening to further work on the topic using appropriate resources and datasets since none were available during the conception of this paper. Adding yet another approach to the ones mentioned in the survey [18] on the integration of language and vision, with a complete $TransferLearning$ based fusion between the two, summarized in:

- Using the BERT [5] embeddings for the IMDB-Dataset [16] to create an IMDB-Image Dataset using the method described in [19] with a t-SNE [24] backbone.
- Analyzing Domain Shifts between the Source (ImageNet [4]) and Target (IMDB-Image) datasets, and avoiding such problems with feature normalization.
- Using early layers from benchmark Vision Models [14] [15] [21] [9] [27] which are trained on a huge ImageNet [4] dataset as feature extractors in a common architecture between five models.
- Training five models containing pre-trained layers on a Vision Dataset(ImageNet [4]) on the IMDB-Image dataset.

In this work, a new approach to classifying textual data is proposed, based on the transformation of text embeddings obtained from the last six layers of BERT [5] into images using t-SNE feature projection. Training five models containing pre-trained layers from Vision Models trained on ImageNet [4], on a generated IMDB-Image Dataset. The contributions of this paper are stated as follows:

- Exchanging knowledge between language and vision models through transfer learning and data transformations.
- Generating an image dataset for IMDB textual dataset, analyzing data domains between source and target datasets (ImageNet [4] and IMDB-Images), avoiding domain shifts with pixel normalization.
- Harnessing the pre-training of Vision models on a large image dataset in text classification, and achieving acceptable and promising results.

II. Related Work

A. Transfer Learning

The modern learning paradigm for most Vision models, is mostly based on extracting features from the ImageNet dataset [4], then finetuning certain layers on a new smaller dataset concerned with a new task. The concept of $TransferLearning$ evolved from when it was first introduced by Stevo Bozinovski and Ante Fulgosi [1] [2]. Due to the revolution in hardware allowing training on very large datasets, all training paradigms nowadays are bound to transfer knowledge obtained from a large dataset to solve a target task with a target dataset. Authors in [33] provide a complete survey on $TransferLearning$, its applications, types, methods, and challenges. Yosinski et al. [28] conducted a detailed study on what kind of features different layers in a Neural Network learn about the source dataset, and how $TransferLearning$ can make the most of the acquired knowledge to solve a target task.

B. Transfer Learning in Language and Vision

Transfer Learning in Computer Vision and Natural Language Processing as two separate sub-fields, is the current standard, as it allows large pre-trained models to be used in multiple tasks after acquiring a certain understanding of features in a large dataset.

1) Transfer Learning in Language: BERT [5] representations and Fine-Tuning, allowed for state-of-the-art results in tasks such as Text Classification, following different approaches mentioned in [22], commonly using the $[CLS]$ representations from certain layers as an input to a classifier, freezing the BERT model or finetuning it depending on the task and size of task dataset. The current approaches mentioned in [22] achieved very low error rates. $TransferLearning$ in this case, is mostly learning contextualized representations for the input text, in a self-supervised manner, where the generated embeddings can be directly used as an input to another model to solve a target task.

2) Transfer Learning in Vision: Vision Models, more specifically [14] [15] [21] [9] [27] are pre-trained on ImageNet [4]. Considered benchmarks for image classification, they achieved very high accuracies for smaller datasets through $TransferLearning$ and $DomainAdaptation$ [3]. Being pre-trained on a very large dataset (ImageNet [4]), the features they can extract are mostly present in most datasets. Finetuning them on a new dataset for a new task, gives optimal results, since after further training both general and task specific features are visible to the finetuned models as suggested and discussed by Yosinski et al. [28].

C. Text Classification using Convolutional Neural Networks

The work conducted by Yoon Kim [12] suggested a CNN based architecture with multiple filter widths and feature maps, and a fully connected layer, using padding to fix the input size to a vector with length d. A similar approach is taken in [31], where sentence classification is achieved, using $Word2Vec$ [17] embeddings, 3 filter region sizes, with 2 filters for each region size, which generates 6 variable size feature maps, forming a feature vector after concatenation, then fed into a Softmax layer. The mentioned approach, treats text embeddings as fixed length inputs, with no regard to which order the tokens in the sentence appear, and no consideration of the geometrical abilities and nature of kernels in a Convolutional Neural Network.

III. Preliminaries

A. IMDB Dataset

IMDB [16] is a Polarity Dataset for Sentiment Analysis or Text Classification in broader terms, it contains 50000 Sentences and their binary class labels, being either "Positive" or "Negative", IMDB is a relatively small dataset that provides a level of flexibility and suitable testing for the study this paper is concerned with, due to the computational resources both Data Generation and Training of different models require.

B. BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [5], is a language representation model with the Transformer [25] as its building block, pre-trained on very large unlabelled textual data for two main tasks: Masked Language Modelling, where the model is required to predict words intentionally masked in a multi-layered context as it was mentioned in the paper, The Second Task is Next Sentence Prediction, also in a self-supervised manner, BERT is trained to classify a sentence as "Next" or "Not Next" for a previous sentence as input. BERT pre-training provides an understanding of the possible relationships, between two sentences especially when being trained on the entire Wikipedia English Corpus and the Books-Corpus [32]. The Attention mechanism heavily used in all layers of BERT produces extremely semantic, and context-sensitive representations, where a token might have many representations depending on the context it is being used in. For the sake of this study, A pre-trained BERT model provided by HuggingFace [26] is used, containing twelve layers,

768 hidden-size (Each Layer produces a 768 sized vector for each word), 12 attention heads, and 110M parameters.[1]

C. DeepInsight

Sharma et al in [19], proposed a methodology to transform non-image data to an image, using t-SNE [24] or K-PCA [6] to project the transpose of a dataset creating a set of features related on a 2D plane according to their similarity, re-transposing the set to obtain the original set size with [NxNx3] images as new samples. The method was heavily used in Cancer Detection and related medical applications.

D. t-SNE

t-SNE [24] (t-distributed Stochastic Neighbor Embedding), is a similarity measuring and dimensionality reduction technique, developed in 2008 by Geoffrey Hinton and Laurens Van Der Maaten. What separates t-SNE [24] from classical dimensionality reduction techniques, is that it is a non-linear method, meaning that datasets with a very large number of dimensions can be easily viewed or projected on a 2D or 3D dimensional space, even when features are not related linearly. The mapping from the high dimensional space to the lower dimensional space (2D or 3D), happens according to the similarity between features and data points, where samples with similar features are clustered together. The similarity between two points is computed as probabilities based on $Euclidean\ Distances$ between pairs of data points.

IV. Method

In this section, the steps taken to generate the dataset used in our experimental part are explained. Going through obtaining BERT-embeddings from the pre-trained BERT [5] model for the original IMDB dataset [16], transforming the embeddings into images, and visualization of some obtained IMDB-Image Dataset samples, and normalizing the pixel space. Along with an analysis of $Source$ and $Target$ domains for the $TransferLearning$ conducted in this paper, and the architectures trained on the generated IMDB-Image dataset, providing sample feature maps produced by the pre-trained layers from [9] [15] [21].

A. Generating the IMDB-Image Dataset

1) BERT Embeddings Generation: A very special feature of BERT is the [CLS] token that is added at the beginning of a sentence embedding at each layer output, which indicates the beginning of a sentence and is also a unique Sentence Representation for classification purposes. Since our procedure requires a high Dimensional Space because we attempt to obtain images after a t-SNE [24] projection of different features representing each input from IMDB [16], as it will be further discussed in following sections. The study conducted in [11], demonstrates the semantic nature of the output from the higher layers of BERT, thus, the [CLS] embeddings from the last six layers are concatenated for each input sentence from the IMDB-Dataset [16], giving a [6x768] sized vector for each input sample as Fig.1 depicts.

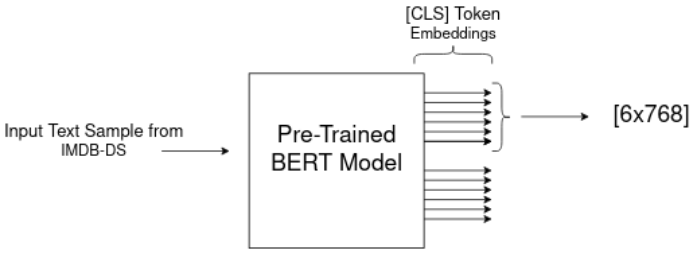

Fig. 1. IMDB [CLS] Embeddings from the Last Six layers of BERT, where the input to the pre-trained model is indexes representing each word in a sentence, outputting a fixed [768] sized vector from each layer. The outputs of $[CLS]$ tokens from the last six layers are stacked into a [6X768] sized embedding for each text sample.

2) Transforming BERT Embeddings into Images: After stacking 6 vectors with 768 features for each sample in the original IMDB [16] dataset, our dataset is now of shape [50000, 4608]. Following the pipeline introduced in [19], We have $n = 50000$ samples, with $d = 4608$ features for each sample, thus our dataset can be defined as $D = \{x_1, x_2, ..., x_n\}$, where each feature vector x is defined as $x = \{f_1, f_2, ..., f_d\}$, our feature set is then defined as $F = \{f_1, f_2, ..., f_d\}$, where each feature f has n dimensions. In short terms $F = D^T$, and transposing the dataset allows for features to be treated as elements that can be related on a 2D plane according to similarity measuring using t-SNE [24]. The obtained 2D plane demonstrated in Fig.2, represents the location of features, and a Convex Hull algorithm is used to isolate the rectangle containing all the points as depicted in Fig.2, the rectangle is then rotated to obtain a horizontal matrix containing Cartesian coordinates for the pixels.

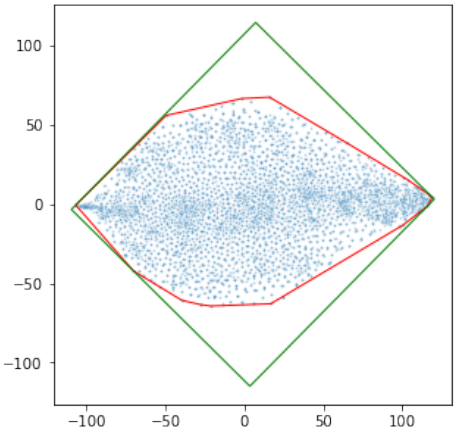

Fig. 2. Feature locations represented by blue points on a 2D plane, as described above. The green rectangle represents the smallest rectangle containing all points and is obtained using a Convex Hull algorithm. We can observe a size of [50x50] is obtained for the rectangle which is then rotated to obtain a horizontal image, ready for use in a Convolutional Neural Network.

[1]more information on the BERT model used is at: https://huggingface.co/transformers/pre-trained_models.html

Since the frame is limited by image size, the points representing feature locations can have more than 1 feature per location, therefore, during mapping features to their locations, averaging is required to avoid confusion which may lead to noisy pixels. Each feature is then mapped to its location, averaging features which fall on the same point or location on the 2D plane. Respecting our hardware capacity and to avoid excessive overlapping of features on the same location, a size of [50x50] for our pixel frames is chosen.

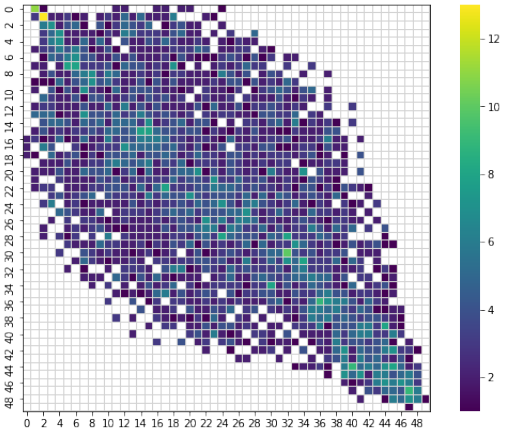

Fig. 3. A Density Matrix, after rotating the Convex containing all the points representing locations projected on a 2D plane with respect to their Cartesian Coordinates obtained from t-SNE [24]. Showing regions that are more dense, due to features being mapped to the same locations causing overlapping. We can observe that our new IMDB-Image Dataset has a certain geometrical distribution of features, with clear edges and blobs.

Fig.3 is an experimental result of applying t-SNE [24] to our [50000,4608] shaped dataset. The Density Matrix reflects the actual distribution of features in the new IMDB-Image Dataset, with an obvious concentration of features on the diagonal, this is due to the overlapping of features if more than one feature is projected with the same Cartesian coordinates on the 2D plane. The next step is to map feature values to their Cartesian coordinates, and averaging feature values that share the same coordinates.

3) Image Data Visualization: After running the method with a t-SNE [24] backbone on our [50000,6x768] IMDB-bert dataset, we obtain gray-scale images with height=50, width=50 and 3 channels having the same values for each pixel, most probably caused by the unnatural data-source which is the pre-trained BERT model which is used to obtain all the values in the [6x768] sized vectors, theoretically forcing the same regularities and irregularities between all samples. Fig.4 demonstrates the Image-Embeddings for the first three IMDB input-samples.
Text Samples for Images in Fig.4:

- "One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. The..."
- "A wonderful little production.

The filming technique is very unassuming- very old-time-B..."
- "I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air con..."

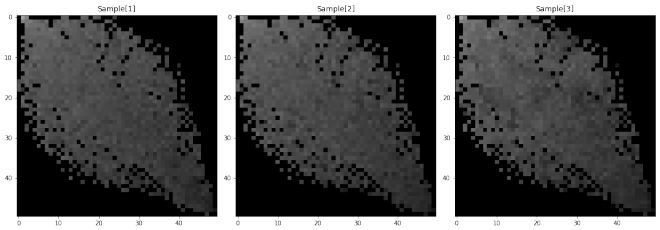

Fig. 4. Image Embeddings for the three first IMDB samples, each image is a projection of a [6x768] sized embedding obtained from BERT for each sample in the IMDB [16] dataset (The text samples above respectively). A new IMDB-Images dataset is generated in the same manner, giving 50000 images, each one representing a text sample in 50000 from the original IMDB [16] dataset.

B. Data Domains and Transfer Learning

Our generated IMDB-Image dataset and the original dataset (ImageNet [4]) on which the models used in this work are trained, are extremely different as depicted in Fig.5, which according to [3] causes a distribution mismatch and domain shift problems to the classifiers. Generalization across domains is extremely affected by the nature of domains and the style of the data especially in Visual Understanding related tasks.

According to [33] a domain D is composed of a feature space χ and a marginal distribution $P(X)$ formulated as:

$$D = \chi, P(X) \tag{1}$$

where X is an instance set which is defined as:

$$X = x|xi \in \chi, i = 1, ..., n$$

The Task T is composed of a label space Y and a decision function f, meaning:

$$T = (Y, f) \tag{2}$$

where f is learned explicitly via training data samples.

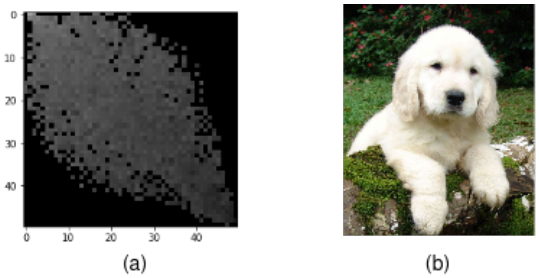

Fig. 5. (a): An image sample from the generated IMDB-Images dataset, showing a gray scale distribution with weak edges and no obvious geometrical features or shapes, (b): An image of a Dog from the $Source$ Dataset (ImageNet [4]), an RGB iamge with clear geometrical features and shapes. This figure depicts the differences between the $Source$ and $Target$ datasets, and suggests inevitable domain shifts between the two.

1) Data Domains: fig.5 shows obvious differences in data style in the source and target datasets, which according to [23] and [20] and further discussed in [30] causes a distribution mismatch and domain shifts, due to the differences between the two domains like channels, colors, background, lighting, etc. This is a major problem in Transfer Learning between image datasets. Survey [3] gives a broad overview of the recent approaches and methods developed for Transfer learning to overcome the mentioned issues through Domain Adaptation.

2) Transfer Learning: For successful Transfer Learning to be achieved, an architecture should be able to adapt the Target Domain D_T to Source Domain D_S, which are IMDB-images and ImageNet [4] respectively in our case. One special technique for domain adaptation so far proposes specific training for domain prediction [7]. Limited by compute power and extreme domain shifts, in this work, another approach is taken. In [28], Jason Yosinski et al, stress on the different kinds of features different zones of layers learn in a neural network, stressing on the generality observed in lower layers and specificity in higher layers, meaning general features like curves, color blobs, and edges are extracted in the lower layers, and task specific features are handled by higher layers. Threatened by the large model sizes in our pre-trained arsenal [14] [15] [21] [9] [27], and the relatively small dataset with only 50000 samples, which could lead to extreme overfitting, the approach taken in this work is to focus on features that both datasets share instead of forcing the model to learn the domains themselves as targets. Since the color channels are clearly going to cause a major domain shift, the focus should be on geometrical features like edges, curves and blobs. In order to have more defined edges, normalization of the entire pixel space is applied, a normalization technique named $Z - Normalization$ [8], adjusting image contrast after moving our input images to a clearer pixel space:

$$\begin{aligned} \mu &= \mathbb{E}_{\mathcal{X}} \in [X], \\ \sigma^2 &= \mathbb{E}_{\mathcal{X}} \in \mathcal{X}[(x-\mu)^2], \\ \hat{x}_i &= \frac{x_i - \mu}{\sigma + \epsilon} \end{aligned} \tag{3}$$

$\mathcal{X}$ is a set of input vectors, μ, σ are the mean and standard deviation of the entire image pixel space, ϵ is a small value to prevent dividing by zero or small denominators.

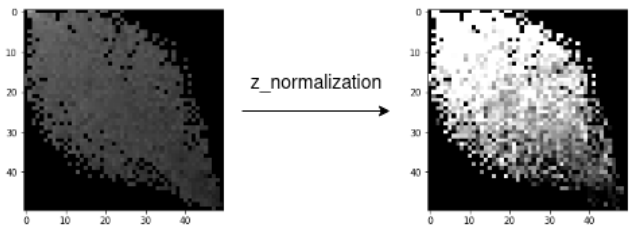

Fig. 6. The image to the left, represents a raw image from the generated IMDB-Images dataset. The image to the right, is the same as the one to its left after $Z - Normalization$, showing stronger edges and more defined geometrical shapes like blobs.

As fig.6 shows, the mentioned normalization technique prevails in creating more distinguishable feature zones in one image, due to the new mean which is close to 0.0 and standard deviation nearing 1.0, which imposes a standard normal distribution on the features, removing noisy regions and enhancing outlying pixels.

C. Architectures used

Since our IMDB-Image dataset is very small compared to the source ImageNet dataset [4], the Convolutional Feature Extractors are sliced from their original pre-trained models, and stacked to a Convolutional Auto-Encoder with randomly initialized parameters, followed by a Dense (Linear) Classifier. A common approach would be to freeze the pre-trained feature extractors, which was followed in this paper to avoid any overfitting problems.

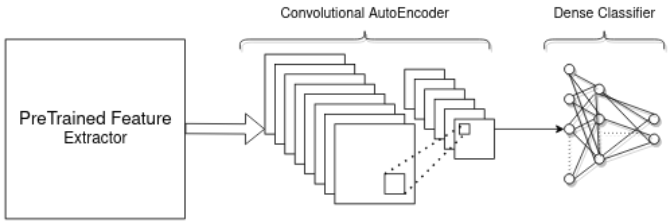

Fig. 7. Main Architecture Used, the pre-trained block represents early layers from five pre-trained Vision Models [14] [15] [21] [9] [27], outputting the input to a Convolutional Auto-encoder, stacked to a Dense Classifier (3 Linear fully connected layers)

As it is depicted in fig.7, the most important part of the architecture is the pre-trained feature extractor. In this paper, for the sake of comparison and confirmation, early layers from five pre-trained models were used as feature extractors, followed by the exact same Conv-AE (Convolutional AutoEncoder) and Dense classifier to ensure fairness in results. The detailed architectures of the pre-trained models are outside the scope, of this paper.

1) pre-trained Feature Extractors: [2]

Visualization of feature maps produced by frozen early layers from the models to be mentioned, depicted in fig.8 lead to choosing specific layers from each model to construct fixed feature extractors to be appended to our standard Conv-AE and Dense Classifier as discussed earlier.

- AlexNet: introduced in [14], Using the first two pre-trained Convolutional Layers, outputs 192 feature maps for each input image from the IMDB-image dataset, with fairly distinguishable differences and focus.
- ResNet: A deep residual model from [9], known as wide-resnet50-2, in torchvision terms. Using the first downsampling Convolutional layer and the first residual layer which contains skip connections described in the original paper.
- ResNext: [27], proposes an aggregated version of the previous ResNet, for the feature extractor we need, the first Convolutional layer, and the first Residual layer are used as well.

[2]All pre-trained models can be found at: https://pytorch.org/vision/stable/models.html

- ShuffleNet V2: From [15] the first Convolutional layer followed by Batch normalization, and stage2 mentioned in the paper.
- VGG16: introduced in [21], We only use the first 12 layers, containing 4 Convolutional layers for the feature extractor.

fig.8 shows sample Feature maps from the feature extractors from: resNet [9],. ShuffleNet [15], Vgg16 [21].

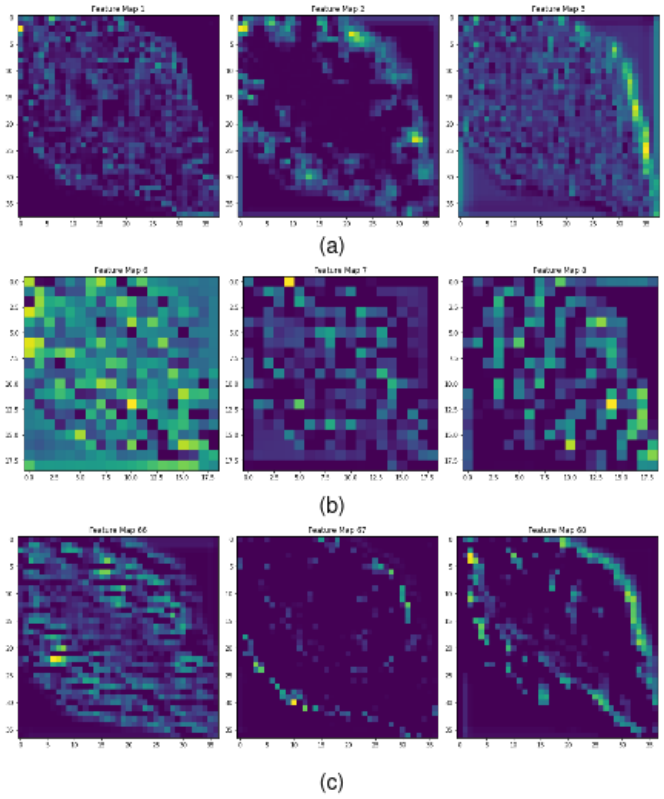

Fig. 8. Feature Maps from early layers of (a) : resNet, (b) : shuffleNet, (c) : VGG16. The pre-trained feature extractors used in our common architecture do in fact output feature maps with defined geometrical features extracted due to the pre-training on the ImageNet [4] dataset. We can observe the adjusted edges and blobs using $Z-Normalization$ being successfully detected by all pre-trained feature extractors, clearest in the depicted three.

In fig.8 we can observe that the pre-trained models are in fact able to extract global features from the new dataset, edges and curves are distinguished by all the pre-trained feature extractors, but clearest in the depicted three. Yielding that [28] stated a concrete study of the transfer-ability of pre-trained models to other datasets and tasks, requiring a considerate treatment of the nature of the features learned by different layers, and the different natures and domains of the source and target datasets.

V. Experiment

This section discusses the training procedure for the mentioned architecture containing five different pre-trained feature extractors from the Vision Models [14] [15] [21] [9] [27], pre-trained on ImageNet [4], and the experimental results achieved by each one on the generated IMDB-Image dataset.

A. Training

As shown in Fig.7, our five models share a Convolutional AutoEncoder, and three Linear Layers, and differ in the pre-trained feature extractors obtained from 5 different pre-trained vision models [14] [15] [21] [9] [27]. Given the differences in data domains depicted in Fig.5, avoiding any possible dataset and covariate shifts is necessary. Using $ReLU$ to keep the same activation patterns within layers of our architecture, given that all 5 pre-trained feature extractors contain $ReLU$ activations. Batch Normalization [10] is applied after each convolutional layer in our Convolutional AutoEncoder block, since our pre-trained feature extractors are trained with 1000 classes on the label side, and our classification task only has 2 classes, this causes an Internal Covariate Shift, representing the change in the distributions of internal nodes of our network as mentioned in [10]. Adam Optimizer [13], is used for all 5 models, to ensure fairness in comparisons, and a fixed batch size of 32. Our IMDB-Image dataset contains 50000 images samples, we split the dataset into a 40000 train and a 10000 validation samples. Differential learning rates are used for our Adam [13] optimizer, while keeping the pre-trained feature extractors frozen during training, since our dataset is very small compared to the $Source$ Dataset (imageNet [4])

An NVIDIA GEFORCE GTX 1060 GPU was used for the entire procedure.

B. Experimental Results

As mentioned in the previous subsection, the different models were trained, with different learning rates, but yet reached very close Validation Accuracies. Higher learning rates caused all models to stagnate and converge very fast to local optimal points, which prevented the models to further learn features necessary to distinguish between different variations contained in each image representing the text samples in the original IMDB-dataset [16].

The following table depicts the number of feature maps outputted by each Feature Extractor along with different Learning Rates used for each one, and the corresponding achieved Validation Accuracies. Fig.9 shows the progress of the validation performance of the five models during training. TABLE 1 and Fig.9 both emphasize the very close results obtained from training all five models on the same IMDB-Image dataset.

Freezing the pre-trained Feature Extractors allows for the complexity of the model as a whole for the five variations to drop, avoiding any chance of overfitting due to the gray scale nature of our IMDB-Image dataset and the RGB channel space of the $Source$ Dataset (ImageNet [4]). The experimental analysis conducted in this paper, is explicitly dedicated to the results obtained by our method and architectures. No comparison with benchmarks or state-of-the-art results in either Image or Text Classification is necessary, since our method is explicitly probing the $Text$ to $Image\ Transformation$ approach described throughout this work.

Given that the pre-trained Feature Extractors, followed by the exact same Convolutional AutoEncoder and Dense Classifier, are trained on the same generated IMDB-Image dataset, and

Feature Ext	Nbr of FM's	CAE LR	LC LR	Val Acc
AlexNet [14]	192	0.00001	0.0005	0.81 (±0.01)
ResNet [9]	256	0.00005	0.0001	0.82 (±0.01)
ResNext [27]	256	0.00005	0.001	0.82 (±0.01)
ShuffleNet [15]	116	0.0005	0.001	0.81 (±0.01)
VGG16 [21]	256	0.00005	0.001	0.81 (±0.01)

TABLE I
NBR OF FM'S (NUMBER OF FEATURE-MAPS), LEARNING RATES FOR CONV-AE (CONVOLUTIONAL AUTOENCODER AND LC (LINEAR CLASSIFIER) AND VAL ACC (VALIDATION ACCURACY) FOR EACH MODEL DEFINED BY ITS FEATURE EXT (EXTRACTOR)

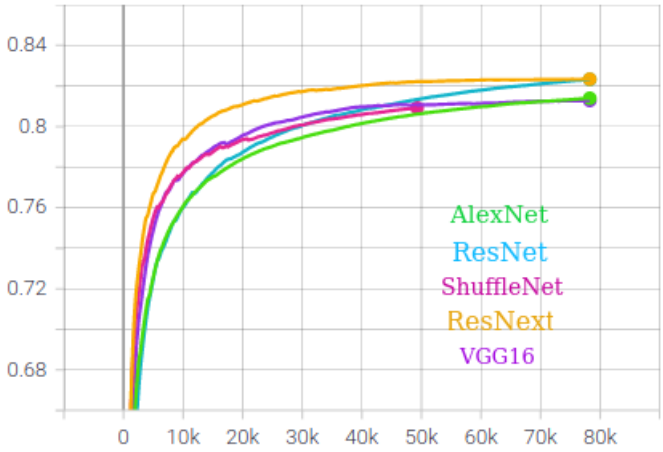

Fig. 9. Validation Accuracy for Five Models during Training process VS. Steps(10 Epochs), where each graph is identified by the model from which its feature extractor is taken. Showing relatively close learning graphs.

observing the experimental results obtained in TABLE 1 and Fig.9, suggests the following:

- The normalized IMDB-Image Dataset, is still not fully avoiding domain shifts due to its raw gray-scale nature.
- The different number of features outputted by each feature extractor, and the almost identical Validation Accuracies, suggests that some feature maps are duplicated, also due to the gray scale nature of the IMDB-Image dataset.
- We can clearly see in Fig.9 that the five models learned at different rates, yet reached close Validation results, this can imply that the models do indeed vary in complexity and generalization abilities, yet limited by dataset size, forcing a maximum validation performance which is almost achieved by all of them.
- The general features learned from the early layers used as feature extractors are extremely similar as suggested in [28], yielding similar results even with different learning rates, since the feature maps play the role of fixed representations for the same dataset.
- Although the Validation Results accomplished do not rise to the State-Of-The-Art in Text Classification, they are still promising given the circumstances and the complexity of the methodology underwent to obtain them.
- The main goal of the paper is achieved, being the classification of text after transforming it into images, with Vision Models pre-trained on ImageNet [4].

As mentioned above, our work is by no means an attempt to reach state-of-the-art results in either $Image\ Classification$ nor $Text\ Classification$. Hence, the only comparison conducted in our experiment, measures how different pre-trained models with different architectures vary in the way they extract feature maps for the newly generated IMDB-Image Dataset.

VI. DATA AND CODE AVAILABILITY

Both the generated IMDB-image dataset, the code for:

- BERT Embedding, Data Transformation and loading.
- Different architectures and training scripts using PyTorch.
- Reproducible paradigm with commented and explained steps.

Are available and ready to be shared.
After initial replies from the conference reviewers, the available dataset and entire code with explained steps for the whole procedure described in this paper will be posted and shared, and links will be added to future versions, with respect to the review period.

VII. CONCLUSION

In this paper a new approach to Sentiment Analysis via Supervised Learning is suggested, using Transfer Learning of pre-trained Vision Models on an Image Dataset (ImageNet [4]), to a textual polarity Dataset (IMDB [16]) embedded using a pre-trained BERT [5] model, where text embeddings are transformed into images using t-SNE [24] feature similarity measuring (Inspired from the work done in DeepInsight [19]), and pre-trained Vision models, are used as feature extractors for a much smaller Neural Classifier to learn sentiment labels. The contributions of our work, is mainly the generation of a new dataset representing textual data from the IMDB [16], avoiding possible domain shifts with pixel normalization, and the expansion to possible future applications using larger datasets and resources. Our results are not compared to the state-of-the-art results because of the unfairness due to $DomainShifts$ and hardware requirements for the generation of an Image Dataset for a larger Textual Dataset. Future work aims to further normalize the discussed approach, as it gives promising results for a small dataset, opening a new challenge for the fusion of Language and Vision via Transfer Learning and Data transformation.

References

[1] S Bozinovski and A Fulgosi. The influence of pattern similarity and transfer learning upon the training of a base perceptron b2.(original in croatian). In *Proceedings of the Symposium Informatica*, pages 3–121.

[2] Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

[3] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Mingtao Ding, Zheng Tian, and Haixia Xu. Adaptive kernel principal component analysis. *Signal processing*, 90(5):1542–1553, 2010.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[8] Dina Q Goldin and Paris C Kanellakis. On similarity queries for time-series data: constraint specification and implementation. In *International Conference on Principles and Practice of Constraint Programming*, pages 137–153. Springer, 1995.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[11] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? 2019.

[12] Yoon Kim. Convolutional neural networks for sentence classification. corr abs/1408.5882 (2014). *arXiv preprint arXiv:1408.5882*, 2014.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[15] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[16] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[18] Aditya Mogadala, Marimuthu Kalimuthu, and Dietrich Klakow. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *arXiv preprint arXiv:1907.09358*, 2019.

[19] Alok Sharma, Edwin Vans, Daichi Shigemizu, Keith A Boroevich, and Tatsuhiko Tsunoda. Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific reports*, 9(1):1–7, 2019.

[20] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[23] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[24] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[27] Saining Xie, Ross B Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. corr abs/1611.05431 (2016). *arXiv preprint arXiv:1611.05431*, 2016.

[28] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[29] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *arXiv preprint arXiv:2104.11892*, 2021.

[30] Lei Zhang and Xinbo Gao. Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687*, 2019.

[31] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[32] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[33] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.