


Article

EmmDocClassifier: Efficient Multimodal Document Image Classifier for Scarce Data

Shrinidhi Kanchi ¹, Alain Pagani ³, Hamam Mokayed ⁴, Marcus Liwicki ⁴, Didier Stricker ^{1,3} and Muhammad Zeshan Afzal ^{1,2,3*} 

- ¹ Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; kanchi@rhrk.uni-kl.de (S.K.);
² Mindgarage, Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
³ German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany; alain.pagani@dfki.de; didier.stricker@dfki.de (D.S.)
⁴ Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden; hamam.mokayed@ltu.se (H.M.); marcus.liwicki@ltu.se (M.L.)
 * Correspondence: muhammad_zeshan.afzal@dfki.de (M.Z.A.)

Abstract: Document classification is one of the most critical steps in the document analysis pipeline. There are two types of approaches for document classification, known as image-based and multimodal approaches. The image-based document classification approaches are solely based on the inherent visual cues of the document images. In contrast, the multimodal approach co-learns the visual and textual features, and it has proved to be more effective. Nonetheless, these approaches require a huge amount of data. This paper presents a novel approach for document classification that works with a small amount of data and outperforms other approaches. The proposed approach incorporates a hierarchical attention network (HAN) for the textual stream and the EfficientNet-B0 for the image stream. The hierarchical attention network in the textual stream uses the dynamic word embedding through fine-tuned BERT. HAN incorporates both the word level and sentence level features. While the earlier approaches rely on training on a large corpus (RVL-CDIP), we show that our approach works with a small amount of data (Tobacco-3482). To this end, we trained the neural network at Tobacco-3428 from scratch. Thereby, we outperform state-of-the-art by obtaining an accuracy of 90.3%. This results in a relative error reduction rate of 7.9%.

Keywords: BERT, Document Image Classification, EfficientNet, fine-tuned BERT, Hierarchical Attention Networks, Multimodal, RVL-CDIP, Two-stream, Tobacco-3482



Citation: Kanchi, S.; Pagani, A.; Mokayed, H.; Liwicki, M.; Stricker, D.; Afzal, M.Z. EmmDocClassifier: Efficient Multimodal Document Image Classifier for Scarce Data. *Preprints* **2021**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Documents play a pivotal role in all of the fields of business communication and record-keeping [1]. Automatic information extraction from documents is a challenging task [2]. In general, the physical documents are first scanned or photographed before the information extraction process can begin. Document classification is considered an essential task in various Document Image Processing Pipelines (DIPP). The classification of documents into different known classes help to improve the overall performance of document processing systems [1]. Consequently, many approaches are proposed for document classification that uses either text content [3–5] or document structure [6–9] to categorize documents into different classes or use both of the modalities [10–13]. There has been much advancement in this area, especially using deep learning methods [6,14,15]. With the advent of AlexNet [16], there has been tremendous growth in the research and performance of Deep neural networks. There has been numerous experiments and researches conducted with different computer vision models like VNet [17], Resnet [18], InceptionNet [19] etc. The documents are treated like conventional images and thus have the models delivered satisfactory results with classification tasks.

Many documents are structurally the same but differ in the textual content. Therefore, these document images convey high-level structural information with their features, but the low-level features that can disambiguate visually similar images remain uninvestigated

for a long time. Various papers investigate the possibility of involving additional features to improve the accuracy like [10], [11] and [13]. These papers obtained state-of-the-art results. They use a powerful OCR engine like [20] to extract the text from the given document images and learn textual features along with visual features. This helps in solving cases where the documents are visually similar.

In this paper, we employ a similar approach, i.e., we use both the visual and textual features. We initially pass the document Images through a visual stream. The previous approaches have used much deeper models such as Resnet [18] and InceptionNet [19]. However, these networks require an enormous amount of training time. To speed up this process, we train the visual stream with EfficientNet[21]. Here, the model is scaled up uniformly with careful balance on the network's depth, width, and resolution. EfficientNet offers various models with different compound coefficients. In particular, EfficientNet-B0 and EfficientNet-B7 have better results speeding up to 6x faster while reducing the network depth up to 12% of the previous sizes. This paper has used EfficientNet-B0, a lighter version of the model, but the results are better than other deep networks such as Resnet-50. As reported in [21], its Top-1 accuracy is nearly 1.5% more than Resnet-50, while it uses nearly five times fewer parameters. Therefore, we used this network that does not only help in improving the accuracy of the visual stream but also reduces the training time considerably.

As discussed above, in some documents like Letters, Memos, and Reports, the contents predominantly have text information. Visually all of them seem similar and also have high inter-class similarity. Therefore, incorporating text features and visual features can result in better performance. Thus we incorporate Hierarchical Attention Network [5] to incorporate text features. The hierarchical Attention Network model performs conventionally better as it splits the text into words and sentences. It gets the information through Attention Model in both levels and combines them hierarchically, thus learning-rich low-level features. This helps immensely in classifying the overlapping type of classes. To enhance this, instead of a static word embedding which is unaware of the context, we replace it with dynamic word embedding, BERT [22]. We initially fine-tune it for our document text corpus, so it learns all the contextual information and maps the words occurring in different contexts as different embedding. This embedding is passed as the embedding layer in the Hierarchical Attention Model. The combination of these two increased the accuracy of the text model by a considerable amount.

The proposed model is better in comparison to other approaches [9–11,23] not only in terms of accuracy but also in terms of efficiency. The previous approaches use huge models in terms of the number of parameters. Training such networks would take a long time and require an extensive dataset. With the EfficientNet model, the network becomes much lighter but provides comparable results. Moreover, HAN provides better textual stream performance. Thus, our contribution in this paper majorly includes improvement in the enhancement of textual stream, reduction in learnable parameters and training time for the visual stream, and improvement in overall performance for Document Image Classification even with very less amount of data.

The rest of the paper is organized as follows: Section 2 walks through prior literary works in the field of Document classification. Next, section 3 narrates textual 3.2 and visual streams 3.1 in detail and, finally, the ensembling of these streams 3.3. Section 4.2 demonstrates the exhaustive overview of the datasets used 4.1, the experimental setup 4.2, implementation details 4.3, and the results 4.4. In the next section 5, we evaluate our results in comparison with the previous results for this task. We also discuss the reasons behind various behaviours of the model. Finally, Section 6 concludes our experiments with the potential future works to elevate the results.

2. Related work

There have been many works done on document image classification. Earlier work extensively focused on extracting text features from the documents and hence was popularly

recognized as text document classification [24–26]. Using CNNs for the document images roots back to the early work proposed by Lecun et al. [27]. In this work, authors employ CNNs for digit recognition. Although it was a shallow network, it gave an excellent result compared to other approaches. However, the work of Afzal et al. [1] of using Alexnet as a classifier model treating documents as images paved the way for increased research on visual features consideration. This work showed that transfer learning could improve the results considerably. Also, in the meantime, Harley et al. [8] had developed an enormous dataset called RVL-CDIP that consisted of 400,000 image documents. Afzal et al. [1] used this dataset to train the model using transfer learning and then used this model to classify the Tobacco-3482 dataset(distinct classes of this dataset are as shown in the Figure 1). All the subsequent works started using this approach. Later, Afzal et al. [7] proposed the use of Resnet-50 as the model, which drastically increased the accuracy of the classification to elevate the number to 91.3%. The numbers truly made a difference to prior work and set a new benchmark.

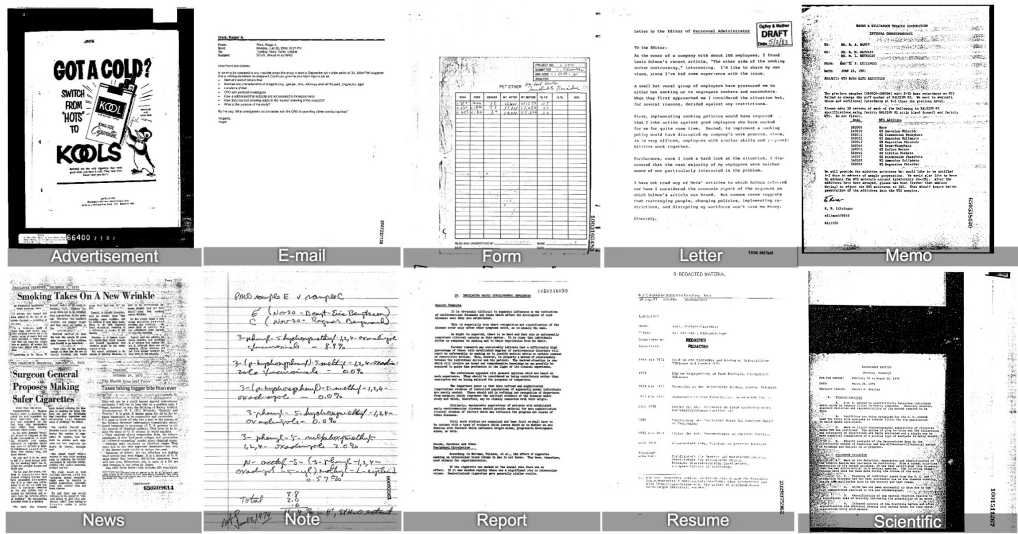


Figure 1. Sample images from each class of Tobacco-3482 dataset. We can see similarities in visual structures for many classes. Also, some documents which have images and hence text plays insignificant roles. In some other, the text is complex or blur for the OCR engine to extract complete or meaningful text from the image. The classes in the image from left to right are: (first row) Advertisement, E-mail, Form, Letter, Memo, (second row) News, Note, Report, Resume, Scientific.

Abuelwafa et al. [28] came up with an unsupervised classification approach. In this work, the author specifies that the model was trained only on the input image and no annotated data. The model was designed to study the visual features of the input image. The authors [28] trained the model on an auxiliary task in which the input was associated with a different label and expanded to multiple images through a data augmentation technique. This method boosted the performance of the model to a greater extent. Koelsch et al. [29] came up with another approach of Real-Time Document Image classification. In their research, the authors propose that the time for training on bigger data for a long time can be reduced using their two-stage approach. In the first stage of the model, the features were extracted using the Deep network. In the latter stage, Extreme Learning Machines(ELMs) were employed to make classification decisions. With this, the relative error was reduced by 25% compared to the previous work of DeepDoc classifier [1] and the training time was significantly reduced to near real-time, and it took only 3.066 seconds to predict close to 2,400 document images.

In another paper by Roy et al. [30], the generalized, compact, and powerful CNNs were used to study the features of the input images, and then Support Vector Machines [31] were employed to combine the individual CNNs. The main contribution from the authors in the method was on the supervised layerwise training of DCNNs in classification tasks

for more accurate weight initialization. Saddami et al. [32] proposed an approach of degradation classification on 3 different datasets for their experiments, namely, Document Image Binarization Contest (DIBCO) [33–40], Persian Heritage Image Binarization Dataset (PHIBD) [41], and private Jawi datasets[42,43] for experimental purposes. Another paper that inspired us to dive into this classification task was the work by Zingaro et al. [44], which focuses on the multimodal side. The authors propose a side-tuning framework for multimodal document classification.

However, all these works missed critical information provided by the text inside and could not categorize highly overlapping document classes. To address this problem, Asim et al [10] came with a new approach in which they extracted both text and image features and ensembled both to give the corresponding class. Their feature ranking algorithm based on the ACC2 method further increased the classification performance. They could achieve an accuracy as high as 95.8%. This pioneering work on the two-stream approach or multimodal ensembling set a new benchmark. Nevertheless, the fact that it relied on text content obtained from OCR-engine that is still not completely accurate made it limited. Although to address this, the authors used multichannel CNN, where one of the channels used Word2Vec [45] initialization, it still could not give a high hit rate. To address this issue, Souhail et al. [11] came up with another approach, where they utilized both static and dynamic word embedding. They used Glove [46] and FastText [47] for static word embedding and BERT for dynamic word embedding. With this experiment, they showed that the result could be improved as BERT focuses on contextualized vector and hence even if OCR failed to identify the words in between, BERT could fill it up based on the context and thus addressing the issue. They got considerably higher numbers for each class compared to earlier approaches like Multichannel CNN. In addition to the InceptionNet used by previous papers like Asim et al [10], Souhail et al [11] also considered using a lighter version of the model to be used for the visual stream, as it would reduce the training time immensely. For the purpose, the authors tried both NasNet [48] Large for actual flow and NasNet [48] mobile for lighter version. This experiment showed that the model could work better even with a lighter version for the classes rich with visual contents. For instance, for the classes like Advertisement, File folder, E-mail, etc., the accuracy of NasNet mobile was almost the same as Nasnet Large. However, it performed badly for the classes like Report, Invoice, etc. The larger version was not the best either, and these classes had to rely on combined textual and visual stream information to get better results. So, the small dip was affordable at the cost of the model's size and the training time.

Inspired by this experiment, Javier et al. [13] came up with another research, in which they employed an even lighter model yet better performing, namely EfficientNet [21]. Also, to speed up the process, they used multiple GPUs and trained using data-parallel. This also helped to increase batch size, leading to better classification performance. Because of this added advantage, the authors reported higher accuracy by employing the model directly on a smaller dataset, Tobacco-3482. The need for training for a longer time on a huge dataset, RVL-CDIP could be circumvented with this. As a result, they achieved accuracy as high as 89.47% compared to earlier approaches by Noce et al. [12] and Audebert et al. [23] that were trained only on Tobacco-3482 and not on RVL-CDIP and had the accuracy of 79.8% and 87.8% respectively.

We propose a novel approach that combines the visual and textual features. While the visual part is inspired by the work of Javier et al. [13], we introduced an improved way of extraction of the textual features that are based on Hierarchical Attention Network. As a result, the combined classifier behaves more intuitively and has significantly improved accuracy.

3. Methodology

This section will explain the model we have proposed and all the details related to the model's training. Although in this paper, we mainly intend to experiment with lighter version models, dynamic word embedding, and hierarchical attentions. For the sake of

The diagram illustrates the proposed HAN framework for document classification. The input image is processed by an OCR Engine and an Image preprocessing module. The OCR Engine outputs text, which is then processed by an Embedding from fine-tuned BERT. The Image preprocessing module outputs a feature map. These two outputs are combined via an Equal concatenation ensemble to feed into the HAN (Hierarchical Attention Network) module. The HAN module consists of multiple layers of attention mechanisms (represented by blue and yellow circles) and is followed by an FC layer (Fully Connected layer) for final classification.

3.1. Visual stream

The network of this particular variant can be seen as shown in Figure 3. This variant has Top-1 accuracy of 77.1% and Top-5 accuracy of 93.3% with only 5.3M learnable parameters instead of 26M parameters of the Resnet50 model. The downsized images after specific preprocessing are passed to this model. The features outputted from EfficientNet-B0 are

then pooled globally before we pass it to a Fully Connected layer, classifying it against 16 classes of RVL-CDIP. In another experiment, we take the images from the Tobacco-3482 dataset, pass them through similar preprocesses and the model structure. Finally, we pass the features from the model to global pooling and Fully Connected layers to classify against 10 classes of Tobacco-3482.

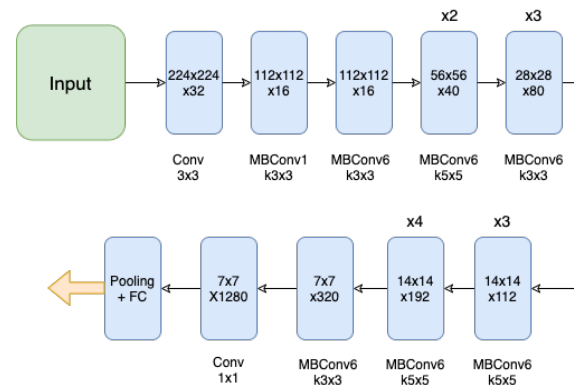


Figure 3. The architecture of EfficientNet [21]. As we can see, it consists of only fewer layers and lesser parameters. However, it delivers 1.5% better top-1 accuracy than the Resnet-50 model, which is nearly 5 times bigger than this model. The 'x' marks on the top indicate number of such layers present.

3.2. Textual stream

In this stream, the textual information is extracted from the documents. Since both the datasets we use for our experiments have only images, the first step is to convert the images to an equivalent textual representation. For this purpose, we employ the Tesseract-OCR. In particular, we use the text extracted from the Tesseract-4 OCR engine. This engine is based on LSTM and simultaneously performs line recognition and character pattern recognition. This generated text contains various symbols and non-alphanumeric characters apart from the punctuation. Unnecessary words/characters that do not add any significance are removed as stopword removal and text preprocessing. Although this helps retain meaningful and weighted words, the text will still contain some meaningless words or malformed words generated from the OCR engine. Moreover, some words are missing in the text due to OCR errors. To address both of the issues, we employ fine-tuned BERT in the next step. Furthermore, It is important to note that in order to address out of vocabulary (OOV) or malformed words, BERT uses a WordPiece tokenization technique. With the help of this, we predict the missing words and generate the dynamic word embedding using the contextual sentence vectors by summing the corresponding word embedding and segment embedding. So, the words from different contexts will have a different entry for embedding, which the model can further utilize. The details regarding BERT are explained below.

BERT: BERT stands for Bidirectional Encoder Representations from Transformers. It encodes the text bi-directionally and requires minimal architectural changes for any downstream natural language processing tasks using a pre-trained transformer encoder. There are two variants of BERT, namely, BERT-Base and BERT-Large. According to [22], the base version consists of 12 layers (transformer blocks), 12 attention heads, and 110 million parameters. The large version consists of 24 layers (transformer blocks), 16 attention heads and, 340 million parameters. In our experiments, we use the base version for fine-tuning and getting the word embedding. Before fine-tuning or fetching word embedding, BERT needs a special text preprocessing.

BERT input sequence consists of text tokens and two other unique tokens, i.e., [CLS] and [SEP]. The input to BERT consists of the concatenation of these tokens. The training of BERT functions is based on predicting an unknown token. To this end, it randomly replaces a token with another special token [MASK] and tries to predict the word from

the context. This step helps BERT to understand the context better in the sentence. This particular token is used only in pretraining and not in other cases.

Every input embedding of BERT is a combination of three embeddings, namely, position embedding, segment embedding, and token embedding, as we can see in Figure 4. Positional embedding is necessary to get the word’s position in the given sentence input. This embedding is essential since it helps BERT to understand the word order. BERT takes sentence pairs as inputs in understanding the context better. Hence it learns a unique embedding for both the sentences to distinguish between them. This is accomplished with segment embedding. As mentioned above, BERT depends on WordPiece token vocabulary. The token embedding is necessary for this. For example, if the word is ‘playing’, this embedding splits the token as ‘play’ and ‘ing’. With these, BERT handles Out-of-Vocabulary(OOV) words.

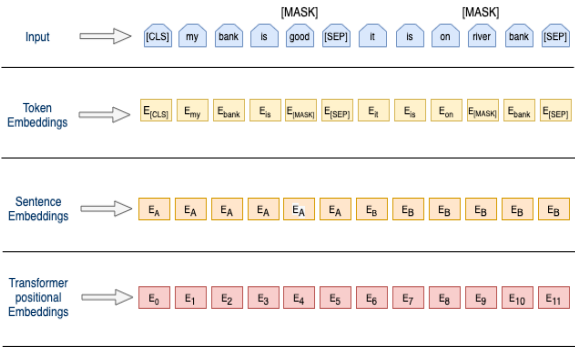


Figure 4. The figure above depicts the typical breakdown of the pair of sentences into three embeddings, Token embedding, Sentence embedding, and Transformer positional Embedding. As shown while pretraining, random tokens are replaced with a special token [MASK]. The interesting factor about BERT is that it can differentiate between the same word ‘bank’ used in different contexts in two different sentences.

After preprocessing, it can be passed on to the pre-trained BERT-Base model, which has 12 layers of transformer encoders. The output from any of these layers can be taken out as the word embedding as they are highly contextualized. However, it is always better to consider top layers since more contextual information is added. With some studies and experiments, it is observed that taking the final four layers and combining them has given better word embedding [50]. As it is clear that there can be different ways to combine the results, and it is entirely data-dependent, we experimented with different strategies and obtained F1 scores for each of them on top frequent tokens from text data.

Strategy	Layer structures	F1 scores
Single Layer	First Layer encoder	91.3
Single Layer	Last Layer encoder	94.2
All layer summation	<div> <div>First Layer encoder</div> <div>Second Layer encoder</div> <div>⋮</div> <div>Penultimate Layer encoder</div> <div>Last Layer encoder</div> </div>	95.8
Final four layer summation	<div> <div>Layer (n-3)</div> <div>Layer (n-2)</div> <div>Layer (n-1)</div> <div>Layer n</div> </div>	96.2
Final four layer concatenation	<div> <div>Layer n</div> <div>Layer (n-1)</div> <div>Layer (n-2)</div> <div>Layer (n-3)</div> </div>	96.1

Figure 5. This Figure depicts different strategies for obtaining better embedding for our problem. Each layer in the figure corresponds to the layer of the BERT-Base, which consists of 12 layers of transformer encoders.

As we can see from Figure 5, both concatenation and summation of the final four layers yield a similar result. We chose the summation approach over concatenation to reduce the embedding dimension. By summing, it would consist of all the information. The only shortcoming of this approach is using a pre-trained BERT-Base model. Although it is trained on a massive dataset and is generic enough to understand, it still precisely lacks knowledge of our corpus. Thus we decided to fine-tune this BERT-Base model on our text corpus. We extract word embedding after the model is trained. Since the context is a prime focus of the BERT transformers, training it for our text helps the model learn context better. The fine-tuned BERT achieved an F1 score of 96.5 for the same set of tokens.

We freeze all the layers of the BERT-Base model except the last four layers and add a fully connected layer to classify the text documents. Even if the text size is small, the broad range BERT model can smartly pick up with the context. Unlike any LSTM or GRU-based model, BERT requires not more than 4 epochs to understand the entire corpus. Once this fine-tuning is done, the weights are frozen, and the fully connected layer is removed. The final embedding is obtained from the vector formed by the summation of the final four layers. This embedding is used for the textual stream model.

HAN: In the next step, we use the HAN model proposed by Yang et al. [5]. This attention network learns the text both from sentence level and word level. By forming the hierarchy of documents, sentences, and words, it tries to learn the linked low-level features. It helps to differentiate highly overlapped classes of documents and also understand at the same time their global contexts. As seen in Figure 6, HAN has 2 hierarchies for word and sentence. For each of these components, it has an encoder and attention. For instance, consider a document D_i , which has n number of sentences in it, and in turn, each sentence has m number of words. Contextualized word vectors are then obtained by passing these words through a fine-tuned BERT embedding. Further, Bidirectional GRU [51] is used to encode these vectors in both directions.

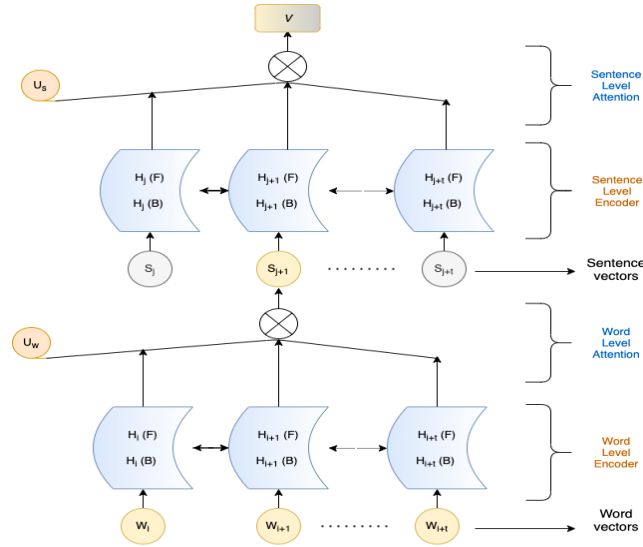


Figure 6. This figure depicts the architecture of Hierarchical Attention Network [5]. First, all the words are passed through Bi-directional GRU(F representing forward, B representing backward) to form the encoder. Then they are then passed to an attention network. These sets of words are then mapped to their corresponding sentence from the input, which goes through a similar process of Bi-directional GRU and Attention. The final vector V then represents the feature vector, which can be used to classify with softmax or combined with the feature vector of the visual stream to form a Two-stream model.

For attention, the encoded-word H_e is fed through MLP with \tanh activation. This is then normalized before taking the weighted sum that eventually forms the sentence s_i . On similar grounds, the sentence level encoding and attention are computed. Finally, bidirectional GRU is applied on each sentence s_{j+1} , and annotation is obtained. This is then fed through MLP with \tanh activation again. After normalizing it, the weighted summation provides us with the document vector. This information can be used to classify the corresponding class when we pass it through fully connected layers. The need to train on huge datasets reduces significantly because of such meticulously formed model and dynamic word embeddings. Thus, in the second experiment, when the model is trained with just the Tobacco-3482 dataset, we get state-of-the-art results.

3.3. Ensembling Visual and Textual Streams

This is the final and important step in our process. Since visual stream performs poorly for some classes and textual stream performs poorly for other classes. The ensembled result is taken from both the streams and then passed on to a convolutional layer. These features are then pooled globally before passing to a fully connected layer to classify them against 10 classes of the Tobacco-3482 dataset. For ensembling, similar to the work of Souhail et al. [11], we tried out both equal concatenation and average ensembling. Therefore, we ensure that both image and text streams have the same feature vector dimension. We empirically found out that equal concatenation worked better, which can mathematically be represented as follows:

$$X_{ens} = X_{image} \oplus X_{text}, \quad X_{image} \in R^d, X_{text} \in R^d \quad (1)$$

where \oplus is the concatenation operator and X_{ens} refers to an ensembled feature of the shape R^{2d} .

4. Experimental Results

4.1. Datasets

For this work, we have used 2 publicly available datasets, namely, RVL-CDIP [8] and Tobacco-3482 [52].

4.1.1. RVL-CDIP

RVL-CDIP stands for Ryerson Vision Lab Complex Document Information Processing. It is a huge dataset with 400,000 grayscale images in it belonging to 16 different classes, *viz.*, Advertisement, Budget, E-mail, File folder, Form, Handwritten, Invoice, Letter, Memos, News article, Presentation, Questionnaire, Resume, Scientific publication, Scientific report, and Specification. Out of 400,000 images, 320,000 images are used as training images, 40,000 images are used as validation, and the remaining 40,000 images are used for testing. This huge dataset which is a subset of IIT-CDIP [53], is another publicly available dataset. This dataset, in turn, is a subset of a legacy Tobacco Document Library [54].

4.1.2. Tobacco-3482

Tobacco-3482 is another publicly available dataset that contains 3482 images belonging to 10 different classes extracted from Legacy Tobacco Document Library [54]. Except for the Note and Report class, all others are already included in the RVL-CDIP dataset. The example images from each of the 10 classes in Tobacco-3482 are shown in Figure 1

4.2. Experimental setup

The pre-processing steps for the experiments can be broadly divided into text pre-processing and image pre-processing as we consider both modalities while training.

As the dataset is originally an image dataset, we need to extract the text from the image before starting with a textual stream. For this purpose, we use Tesseract-OCR as mentioned above. Although this LSTM-based powerful engine can extract the text with reasonable accuracy, the text still contains a lot of misspellings or letters not correctly recognised because of the discrepancies in the source image itself. For instance, from Figure 7 (A) and (B), we can observe a high amount of noise in the source image for OCR-engine to extract the content. Although we can get most content, it will still have many incorrect text or misspellings. Besides, even the correctly extracted text contains a high number of stop-words. To help the text model learn the context easily, we apply some of the Natural Language Processing techniques. In our experiment, we first remove all incorrect characters or symbols. Then, we remove stop words from the cleaned text and retain them with stemmed words. This pre-processing plays a vital role in classifying the text accurately.

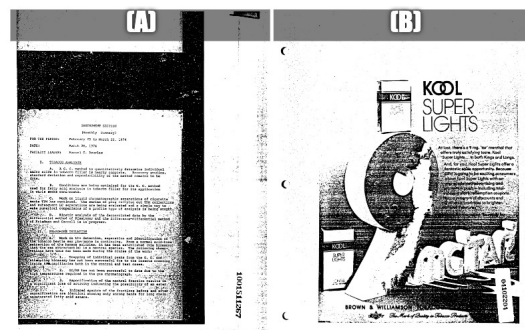


Figure 7. Image samples where the text is blurred or has too much noise leads OCR to extract the text incorrectly. Thus, the text contains incorrect characters or misspellings.

For the visual stream, we use EfficientNet, which accepts a fixed-size input of 384x384. So, as a first step, we downsample all the images (originally in the size of 1000x750) to 384x384. Most of the image pre-processing required for the model is included in standard

deep learning libraries. However, the networks trained on ImageNet dataset requires 3-channel input. Therefore, we convert the grayscale images to RGB images by copying the same content in all channels.

4.3. Implementation details

In this section, we will go through the implementation details. In the first step of the textual stream, the text is passed through fine-tuned BERT. For this purpose, we implement a fine-tuned BERT. We use a maximum of 25 sentences per document and a maximum of 10,000 words to improve performance while not hindering the accuracy. The tokens and segments are stacked together and passed on to the BERT model which is fine-tuned for the text corpus. From the features returned model, the final 4 layers are summed to get the vector used as the embedding for the given word token. The fine-tuned BERT handles the cases for OOV and malformed words.

Once we have the embedding matrix ready from the previous step, we will use that in the embedding layer of the HAN model that learns word level and sentence level features hierarchically with the help of the TimeDistributed function. This whole model is trained on RVL-CDIP in one experiment and Tobacco-3482 in the other. We use RMSProp optimizer with the learning rate of 0.001, β of 0.9 and ϵ of 1.0. The model is trained with a batch size of 100 for 20 epochs. Categorical cross-entropy is used as the loss. We use early stopping criteria and quit if the loss does not change significantly for at least 4 epochs.

For the visual stream, we employ EfficientNet-B0 for classifying the images. it is trained with Adam optimizer with the initial learning rate of 0.001, β_1 0.9, and β_2 0.999. We use the custom learning rate scheduler while training. In a total of 20 epochs of training, we train the first 3 epochs with an initial learning rate and then reduce it to 1e-4 for the next 4 iterations. The rest of the iterations are trained with a learning rate of 1e-5.

Finally, the two-stream combined model is trained with Adam optimiser with the learning rate of 0.001 for 20 epochs and batch size of 100. We use categorical cross-entropy as loss for the classification problem against 10 classes of the Tobacco-3482 dataset.

4.4. Results

This section will discuss the results we have obtained through the two experiments conducted as part of this work. In the first experiment, we first trained the model on RVL-CDIP. On its evaluation, the predictions were as accurate as 95.48%. We then took this model and used transfer learning to test it on the Tobacco-3482 dataset. We achieved an accuracy of 95.7%, which is nearly equal to the state-of-the-art result by Souhail et al. [11] with the value of 96.94%. While the EfficientNet standalone gave the accuracy of around 94.04%, the number was elevated as a combination with the textual stream, which had dynamic word embedding and hierarchical attention networks. The overall accuracy compared to prior approaches are listed below in Table 1

Table 1. Accuracy comparison of the proposed method to the former methods on Tobacco-3482 dataset. Our work surpasses all previous performance compared to only those approaches that do not train on the bigger dataset.

Authors	Imagenet + RVL-CDIP		Imagenet + No RVL-CDIP	
	Image	Both modalities	Image	Both modalities
Kumar et al. (2014) [55]	-	-	43.8	-
Kang et al. (2014) [56]	-	-	65.37	-
Afzal et al. (2015) [1]	-	-	77.6	-
Harley et al. (2015) [8]	89.8	-	79.9	-
Noce et al. (2016) [12]	-	-	-	79.8
Afzal et al. (2017) [7]	91.13	-	-	-
Das et al. (2018) [9]	92.21	-	-	-
Audebert et al. (2019) [23]	-	-	84.5	87.8
Asim et al. (2019) [10]	93.2	95.8	-	-
Souhail et al. (2020) [11]	91.3	96.94	-	-
Javier et al. (2020) [13]	94.04	94.9	85.99	89.47
Our approach	94.04	95.7	86.0	90.3

The focus of this paper lies with the second experiment, in which we circumvent the long training period taken in other approaches. In this experiment, instead of training on the large corpus and then transferring the learning, we directly train on the smaller dataset, Tobacco-3482, for both images and text and then classify the test data. As a result, we get an accuracy of 90.3% which is nearly 1% more than the state-of-the-art [13]. The earlier approach of Javier et al. [13] has an accuracy of 89.47% and [23] has an accuracy of 87.8%.

This considerable improvement in accuracy is due to the proposed pipeline that uses hierarchical attention networks and fine-tuned BERT embedding. As an ablation study, when BERT embedding was replaced by FastText word embedding, the result was poorer than the result of [13]. The experiment by Javier et al. [13] was the other way round where they used a fine-tuned BERT model. When both of these were employed together, we could achieve almost 1% higher accuracy than them.

5. Discussion and Evaluation of the results

In this section, we will evaluate the results in detail. We will walk through some examples where the model behaved differently than expected, for example, incorrectly classified to a different class. We will also discuss possible future directions for improvements. Figure 8 shows some examples of correctly and wrongly classified documents.

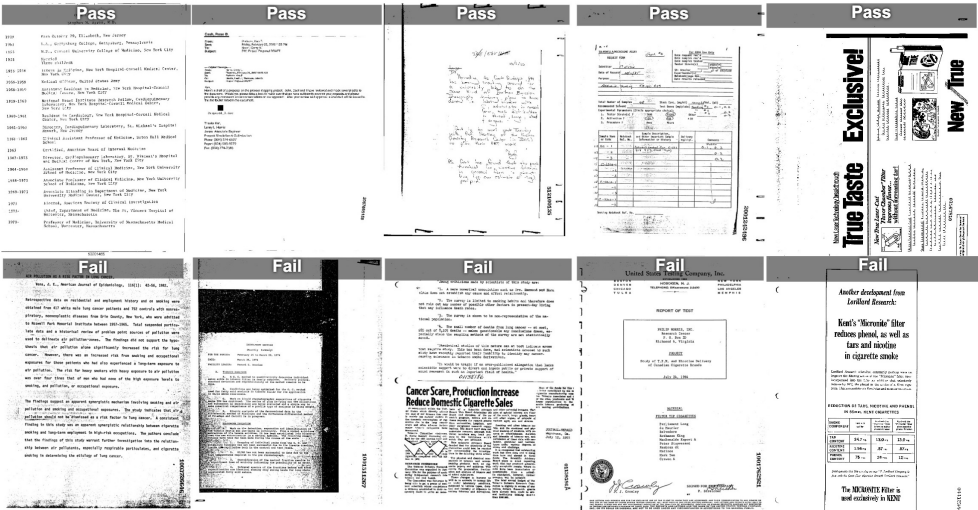


Figure 8. This figure is a collection of 10 different images taken randomly from the dataset, in which 5 of them are successfully classified, while the other 5 were incorrectly classified. The potential reasons for failure in identifying the classes of these input images are discussed in detail in section 5

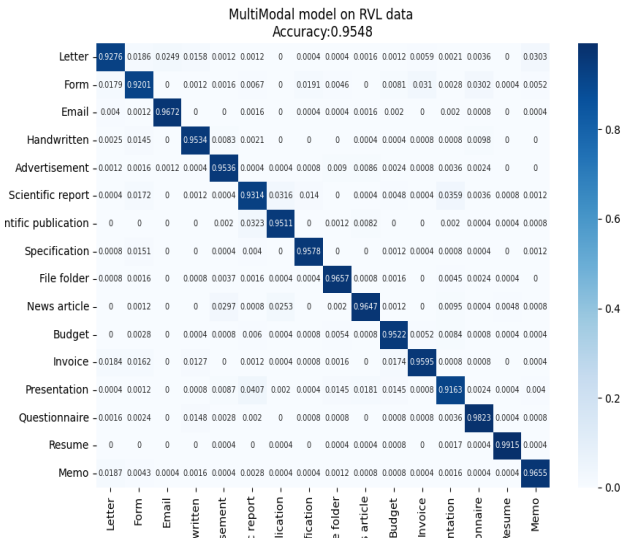


Figure 9. Confusion matrix for the model on RVL-CDIP test data. This evaluation is necessary to perfectly fit the model before transferring this learning to the Tobacco-3482 dataset.

First, as we train the model initially on RVL-CDIP dataset for the first experiment, we set all hyper-parameters. Then, once we receive good results, we proceed with transfer learning. As we can see in Figure 9, although all the classes are classified correctly to a greater extent, some of the classes are confused with other classes ranging from 3 to 4%. This high misclassification is due to more overlapping features between them. For example, the two classes, Scientific Publication and Scientific Report are mutually misclassified in almost 3% of the cases. Also, if we look carefully into a few images of the classes Scientific report and Presentation, we can see many similarities in visual and textual features. That explains the failure for around 4% of the cases in the confusion matrix. Also, if we look at the classes E-mail and Memo, we see that they have minimal overlap with any other classes, which is justified by the precise boundary of the features from other class features.

As depicted in Figure 10, the overall behaviour of the model can be understood while tested on the Tobacco-3482 dataset. Whether or not the model is pre-trained with the RVL-CDIP dataset, the model can easily classify certain classes like E-mail, Form, Memo, etc.

As we can see in Figure 1, classes like E-mail, Form and Memo have apparent visual and textual features completely independent and non-overlapping with other classes. Hence, both visual and textual streams deliver the features that help the classifier categorise them into their corresponding classes.

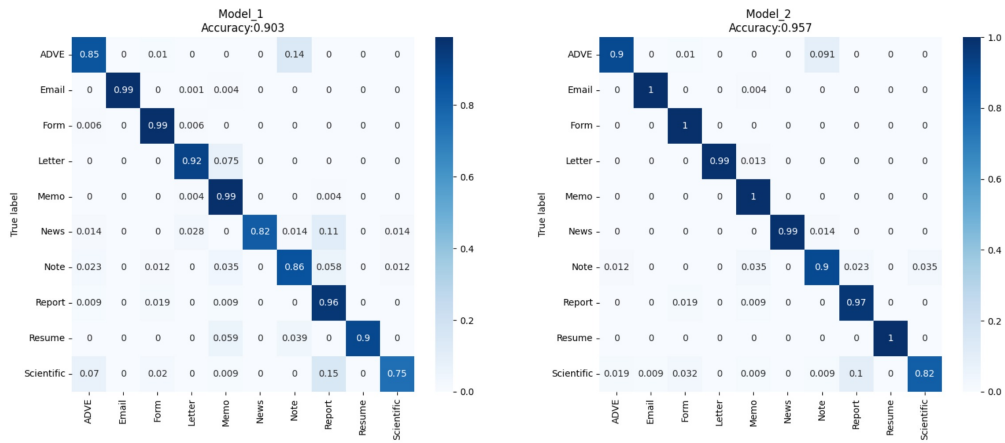


Figure 10. Confusion matrix for the models from two different experiments carried out in this paper. The Model_1 matrix on the left is a model without RVL-CDIP pretraining, instead directly trained and tested on the Tobacco-3482 dataset. The visual stream uses pre-trained ImageNet weights in this model, while the textual stream does not use pre-trained weights. However, before the multimodal training, the BERT embedding is fine-tuned for the Tobacco-3482 dataset. The Model_2 matrix on the right is a model trained on RVL-CDIP prior and then trained and tested on the Tobacco-3482 dataset.

However, this section focuses on those scenarios where the classifier failed to classify the input to its respective class. As we can see from the Confusion Matrix 10, the model with no pretraining on the RVL-CDIP dataset performs better on most of the classes except for News, Note, and Scientific. On a higher level, the reason is that these classes have a high overlapping number of features with other classes. While the visual stream fails to identify these classes, the textual stream fails because of potentially less text or improper text extracted by the OCR engine due to blurred and noisy images.

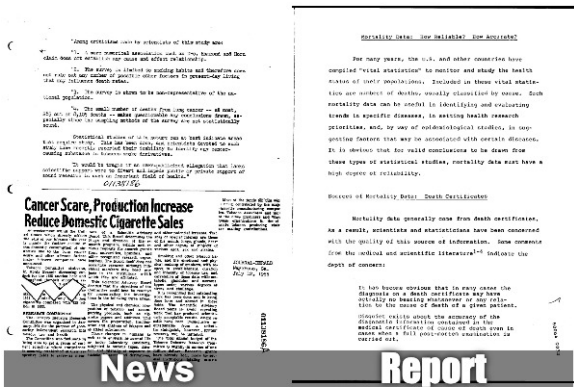


Figure 11. The sample image inputs from two classes News and Report. As we can see from the two images, the image of News matches the image from Report to a greater extent in visual features.

Let us consider an image from the class News, which visually resembles another class Report. For instance, let the input image be the image from News, as shown in the figure 11. The model trained only on the Tobacco-3482 dataset classifies it as Report. Although it contains the strong News features in the lower half of the image, the upper half is similar to the Report. Due to the limited data availability, the model fails to classify it correctly. However, the same image input while passed through a model which is pre-trained on

RVL-CDIP dataset, because of a wide variety of data it has learned upon, the attention network and visual features classify it as News. We can see such subtle differences from the confusion matrix [10] with some other classes as well. The false classification of Scientific for Advertisement, News for Report, Advertisement for News, etc. can be improved with RVL-CDIP pre-training.

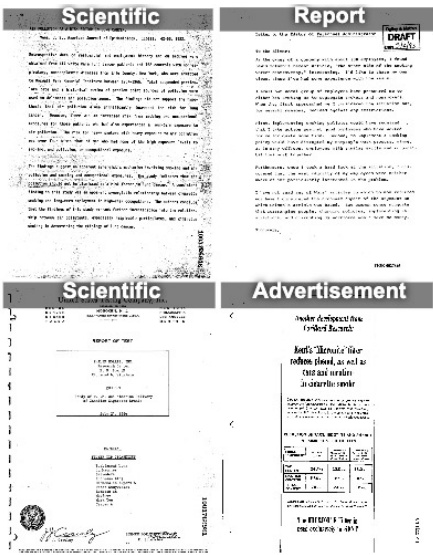


Figure 12. In this set of pictures, we can see the resemblance of the class Scientific to the class Report. In some other cases, although on a lesser amount, it resembles a lot to another class Advertisement.

Overall, for both the models, we can see that the class Scientific is poorly classified. A careful study on this has been done on why and when the classifier fails. It is observed that the classifier failed to identify properly the class Scientific, but no other classes were mixed up with Scientific. During analysis, we found out that there are only a few defining features for the class Scientific. It mostly resembles another class Report with clear-cut features in both textual and visual streams. As we can see in Figure 12 the image from Scientific is very much similar to the one from Report. It is both visually and textually impossible to make a clear decision. Since the features for Report are fixed because of its discipline, even the scientific is confused with being a Report in this case. Furthermore, another class Advertisement resembles some of the images of class Scientific. The samples are depicted in 12 for reference. In addition to overlapping features, another reason for poor performance in the Scientific class is the amount of noise in the images from this class. As we can see from Figure 7, the image (A), which belongs to the class Scientific, has too much noise that has blurred the text in it. As a result, the OCR engine fails to extract the exact text, and thus even the textual stream, which is supposed to identify the class based on attention and contextual information fails to identify the image correctly. One future direction to improve the performance is to augment the data with synthetically generated images that can highlight the salient features.

6. Conclusion

We presented an efficient multimodal neural network EmmDocClassifier for document image classification. We show that the network works reasonably well even with a small amount of data. We attribute this capability of the network to improved textual feature learning that uses Hierarchical Attention Network. In our experiments, we train the proposed network on the Tobacco-3084 dataset from scratch. We obtain an accuracy of 90.3%. We outperform the current state-of-the-art [13] and reduce the relative error by 7.9%. The efficiency of the proposed network is attributed to the EfficientNet-B0 that is used for the feature extraction from the visual stream. The reduction in the size of the network is about 80%. As a result, we reduce both the training and the inference time. While other

approaches use huge networks that are difficult to use on-device training, the proposed network with a little bit of modification is suitable for this purpose. In future, we are planning to improve the results by introducing co-attention between the two modalities. Another interesting future direction is the further reduction in the size of the network.

Author Contributions: writing—original draft preparation, S.K., M.Z.A.; writing—review and editing, S.K., M.Z.A., H.M.; supervision and project administration, M.L., A.P., D.S. All authors have read and agreed to the submitted version of the manuscript.

Funding: Not applicable.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Afzal, M.Z.; Capobianco, S.; Malik, M.I.; Marinai, S.; Breuel, T.M.; Dengel, A.; Liwicki, M. Deepdocclassifier: Document classification with deep convolutional neural network **2015**. pp. 1111–1115.
2. Adnan, K.; Akbar, R. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management* **2019**, *11*, 1847979019890771, [<https://doi.org/10.1177/1847979019890771>]. doi: 10.1177/1847979019890771.
3. Choi, G.; Oh, S.; Kim, H. Improving Document-Level Sentiment Classification Using Importance of Sentences. *Entropy* **2020**, *22*, 1336. doi:10.3390/e22121336.
4. Adhikari, A.; Ram, A.; Tang, R.; Lin, J. Rethinking Complex Neural Network Architectures for Document Classification. 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); Association for Computational Linguistics: Minneapolis, Minnesota, 2019; pp. 4046–4051. doi:10.18653/v1/N19-1408.
5. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical Attention Networks for Document Classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Association for Computational Linguistics: San Diego, California, 2016; pp. 1480–1489. doi: 10.18653/v1/N16-1174.
6. Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; Zhang, M.; Zhou, L. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding, 2021, [[arXiv:cs.CL/2012.14740](https://arxiv.org/abs/2012.14740)].
7. Afzal, M.Z.; Kölsch, A.; Ahmed, S.; Liwicki, M. Cutting the error by half: Investigation of very deep cnn and advanced training strategies for document image classification. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); IEEE, , 2017; Vol. 1, pp. 883–888.
8. Harley, A.W.; Ufkes, A.; Derpanis, K.G. Evaluation of deep convolutional nets for document image classification and retrieval. 2015 13th International Conference on Document Analysis and Recognition (ICDAR); IEEE, , 2015; pp. 991–995.
9. Das, A.; Roy, S.; Bhattacharya, U.; Parui, S.K. Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 3180–3185.
10. Asim, M.N.; Khan, M.U.G.; Malik, M.I.; Razzaque, K.; Dengel, A.; Ahmed, S. Two stream deep network for document image classification. 2019 International Conference on Document Analysis and Recognition (ICDAR); IEEE, , 2019; pp. 1410–1416.
11. Bakkali, S.; Ming, Z.; Coustaty, M.; Rusinol, M. Visual and textual deep feature fusion for document image classification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 562–563.
12. Noce, L.; Gallo, I.; Zamberletti, A.; Calefati, A. Embedded Textual Content for Document Image Classification with Convolutional Neural Networks. Proceedings of the 2016 ACM Symposium

- on Document Engineering; Association for Computing Machinery: New York, NY, USA, 2016; DocEng '16, p. 165–173. doi:10.1145/2960811.2960814.
13. Ferrando, J.; Domínguez, J.L.; Torres, J.; García, R.; García, D.; Garrido, D.; Cortada, J.; Valero, M. Improving Accuracy and Speeding Up Document Image Classification Through Parallel Systems. *Computational Science – ICCS 2020*; Krzhizhanovskaya, V.V.; Závodszy, G.; Lees, M.H.; Dongarra, J.J.; Sloot, P.M.A.; Brissos, S.; Teixeira, J., Eds.; Springer International Publishing: Cham, 2020; pp. 387–400.
 14. Powalski, R.; Łukasz Borchmann.; Jurkiewicz, D.; Dwojak, T.; Pietruszka, M.; Pałka, G. Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer, 2021, [arXiv:cs.CL/2102.09550].
 15. Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M. Layoutlm: Pre-training of text and layout for document image understanding 2020. pp. 1192–1200.
 16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*; Pereira, F.; Burges, C.J.C.; Bottou, L.; Weinberger, K.Q., Eds. Curran Associates, Inc., 2012, Vol. 25, pp. 1097–1105.
 17. de la Fuente Castillo, V.; Díaz-Álvarez, A.; Manso-Callejo, M.Á.; Serradilla Garcia, F. Grammar guided genetic programming for network architecture search and road detection on aerial orthophotography. *Applied Sciences* 2020, 10, 3953.
 18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition, 2016.
 19. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision, 2015, [arXiv:cs.CV/1512.00567].
 20. Kay, A. Tesseract: an open-source optical character recognition engine. *Linux Journal* 2007, 2007, 2.
 21. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020, [arXiv:cs.LG/1905.11946].
 22. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019, [arXiv:cs.CL/1810.04805].
 23. Audebert, N.; Herold, C.; Slimani, K.; Vidal, C. Multimodal deep networks for text and image-based document classification, 2019, [arXiv:cs.CV/1907.06370].
 24. Cavnar, W.B.; Trenkle, J.M. N-Gram-Based Text Categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.
 25. Hoch, R. Using IR Techniques for Text Classification in Document Analysis. *SIGIR '94*; Croft, B.W.; van Rijsbergen, C.J., Eds.; Springer London: London, 1994; pp. 31–40.
 26. Ittner, D.J.; Lewis, D.D.; Ahn, D.D. Text categorization of low quality images. *Symposium on Document Analysis and Information Retrieval*. Citeseer, 1995, pp. 301–315.
 27. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 1998, 86, 2278–2324. doi:10.1109/5.726791.
 28. Abuelwafa, S.; Pedersoli, M.; Cheriet, M. Unsupervised Exemplar-Based Learning for Improved Document Image Classification. *IEEE Access* 2019, 7, 133738–133748. doi:10.1109/ACCESS.2019.2940884.
 29. Kölsch, A.; Afzal, M.Z.; Ebbecke, M.; Liwicki, M. Real-time document image classification using deep CNN and extreme learning machines. 2017 14th IAPR international conference on document analysis and recognition (ICDAR); IEEE, , 2017; Vol. 1, pp. 1318–1323.
 30. Roy, S.; Das, A.; Bhattacharya, U. Generalized stacking of layerwise-trained deep convolutional neural networks for document image classification. 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 2016, pp. 1273–1278.
 31. Hearst, M.A. Support Vector Machines. *IEEE Intelligent Systems* 1998, 13, 18–28. doi: 10.1109/5254.708428.
 32. Saddami, K.; Munadi, K.; Arnia, F. Degradation Classification on Ancient Document Image Based on Deep Neural Networks. 2020 3rd International Conference on Information and Communications Technology (ICOIACT). IEEE, 2020, pp. 405–410.
 33. Gatos, B.; Ntirogiannis, K.; Pratikakis, I. ICDAR 2009 document image binarization contest (DIBCO 2009). 2009 10th International conference on document analysis and recognition; IEEE, , 2009; pp. 1375–1382.
 34. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). 2011 International Conference on Document Analysis and Recognition; , 2011; pp. 1506–1510. doi:10.1109/ICDAR.2011.299.

35. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. H-DIBCO 2010 - Handwritten Document Image Binarization Competition. 2010 12th International Conference on Frontiers in Handwriting Recognition, 2010, pp. 727–732. doi:10.1109/ICFHR.2010.118.
36. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012). 2012 International Conference on Frontiers in Handwriting Recognition, 2012, pp. 817–822. doi:10.1109/ICFHR.2012.216.
37. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICDAR 2013 Document Image Binarization Contest (DIBCO 2013). 2013 12th International Conference on Document Analysis and Recognition; , 2013; pp. 1471–1476. doi:10.1109/ICDAR.2013.219.
38. Ntirogiannis, K.; Gatos, B.; Pratikakis, I. ICFHR2014 Competition on Handwritten Document Image Binarization (H-DIBCO 2014). 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014, pp. 809–813. doi:10.1109/ICFHR.2014.141.
39. Pratikakis, I.; Zagoris, K.; Barlas, G.; Gatos, B. ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016). 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 619–623. doi:10.1109/ICFHR.2016.0118.
40. Pratikakis, I.; Zagoris, K.; Barlas, G.; Gatos, B. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); , 2017; Vol. 01, pp. 1395–1403. doi:10.1109/ICDAR.2017.228.
41. Ayatollahi, S.; Nafchi, H. Persian heritage image binarization competition (PHIBC 2012). 2013, pp. 1–4. doi:10.1109/PRIA.2013.6528442.
42. Saddami, K.; Munadi, K.; Muchallil, S.; Arnia, F. Improved Thresholding Method for Enhancing Jawi Binarization Performance. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR); , 2017; Vol. 01, pp. 1108–1113. doi:10.1109/ICDAR.2017.183.
43. Saddami, K.; Afrah, P.; Mutiawani, V.; Arnia, F. A New Adaptive Thresholding Technique for Binarizing Ancient Document. 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 2018, pp. 57–61. doi:10.1109/INAPR.2018.8627036.
44. Zingaro, S.P.; Lisanti, G.; Gabbrielli, M. Multimodal Side- Tuning for Document Classification. 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 5206–5213. doi: 10.1109/ICPR48806.2021.9413208.
45. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space, 2013, [arXiv:cs.CL/1301.3781].
46. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
47. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* 2016.
48. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition, 2018, [arXiv:cs.CV/1707.07012].
49. Deng, J. W., Dong, R. Socher, L. J., Li, K., Li, and, L., Fei, Fei., Imagenet: A large-scale, hierarchical, image, database., In, IEEE, Conference, on Computer, Vision, and Pattern, Recognition, (CVPR),,, pages, 2009, pp. 248–255.
50. Xiao, H. bert-as-service. <https://github.com/hanxiao/bert-as-service>, 2018.
51. Mangal, S.; Joshi, P.; Modak, R. LSTM vs. GRU vs. Bidirectional RNN for script generation, 2019, [arXiv:cs.CL/1908.04332].
52. Kumar, J.; Ye, P.; Doermann, D. Learning document structure for retrieval and classification. Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012); , 2012; pp. 1558–1561.
53. Lewis, D.; Agam, G.; Argamon, S.; Frieder, O.; Grossman, D.; Heard, J. Building a Test Collection for Complex Document Information Processing. Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; Association for Computing Machinery: New York, NY, USA, 2006; SIGIR '06, p. 665–666. doi: 10.1145/1148170.1148307.
54. The Legacy Tobacco Document Library (LTDL), University of California, San Francisco. <http://legacy.library.ucsf.edu/>. Published: 2007.
55. Kumar, J.; Ye, P.; Doermann, D. Structural Similarity for Document Image Classification and Retrieval. *Pattern Recognition Letters* 2014, 43, 119–126.
56. Kang, L.; Kumar, J.; Ye, P.; Li, Y.; Doermann, D. Convolutional Neural Networks for Document Image Classification. 2014 22nd International Conference on Pattern Recognition 2014, pp. 3168–3172.

