Multimodal deep networks for text and image-based document classification

Nicolas Audebert Catherine Herold Kuider Slimani Cédric Vidal

Quicksign, 38 rue du Sentier, 75002 Paris

{nicolas.audebert,catherine.herold,kuider.slimani,cedric.vidal}@quicksign.com

Résumé

La classification automatique de documents numérisés est importante pour la dématérialisation de documents historiques comme de procédures administratives. De premières approches ont été suggérées en appliquant des réseaux convolutifs aux images de documents en exploitant leur aspect visuel. Toutefois, la précision des classes demandée dans un contexte réel dépend souvent de l'information réellement contenue dans le texte, et pas seulement dans l'image. Nous introduisons un réseau de neurones multimodal capable d'apprendre à partir d'un plongement lexical du texte extrait par reconnaissance de caractères et des caractéristiques visuelles de l'image. Nous démontrons la pertinence expérimentale de cette approche sur les jeux de données Tobacco3482 et RVL-CDIP, sur lesquels nous améliorons les performances d'un modèle image de 3% grâce à l'information sémantique textuelle.

Mots Clef

Classification de documents, apprentissage multimodal, fusion de données.

Abstract

Classification of document images is a critical step for archival of old manuscripts, online subscription and administrative procedures. Computer vision and deep learning have been suggested as a first solution to classify documents based on their visual appearance. However, achieving the fine-grained classification that is required in real-world setting cannot be achieved by visual analysis alone. Often, the relevant information is in the actual text content of the document. We design a multimodal neural network that is able to learn from word embeddings, computed on text extracted by OCR, and from the image. We show that this approach boosts pure image accuracy by 3% on the Tobacco3482 and RVL-CDIP datasets, even without clean text information.

Keywords

Document classification, multimodal learning, data fusion.

1 Introduction

The ubiquity of computers and smartphones has incentivized governments and companies alike to digitize most of their processes. Onboarding new clients, paying taxes and proving one's identity is more and more done through a computer, as the rise of online banking has shown in the last few years. Industrial and public archives are also ongoing serious efforts to digitize their content in an effort for preservation, e.g. for old manuscripts, maps and documents with a historical value. This means that previously physical records, such as forms and identity documents, are now digitized and transferred electronically. In some cases, those records are produced and consumed by fully automated systems that rely on machine-readable formats, such as XML or PDF with text layers. However, most of these digital copies are generated by end-users using whatever mean they have access to, i.e. scanners and cameras, especially from smartphones. For this reason, human operators have remained needed to proofread the documents, extract selected fields, check the records' consistency and ensure that the appropriate files have been submitted. Automation through expert systems and machine learning can help accelerate this process to assist and alleviate the burden of this fastidious work for human workers.

A common task involved in data filing processes is document recognition, on which depends the class-specific rules that command each file. For example, a user might be asked to upload several documents such as a filled subscription form, an ID and a proof-of-residence. In this work, we tackle the document classification task to check that all required files have been sent so that they are filed accordingly.

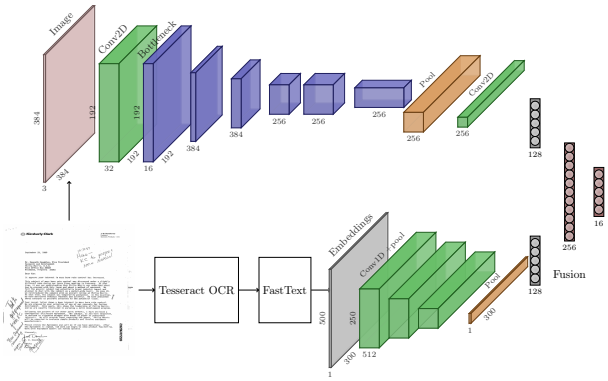
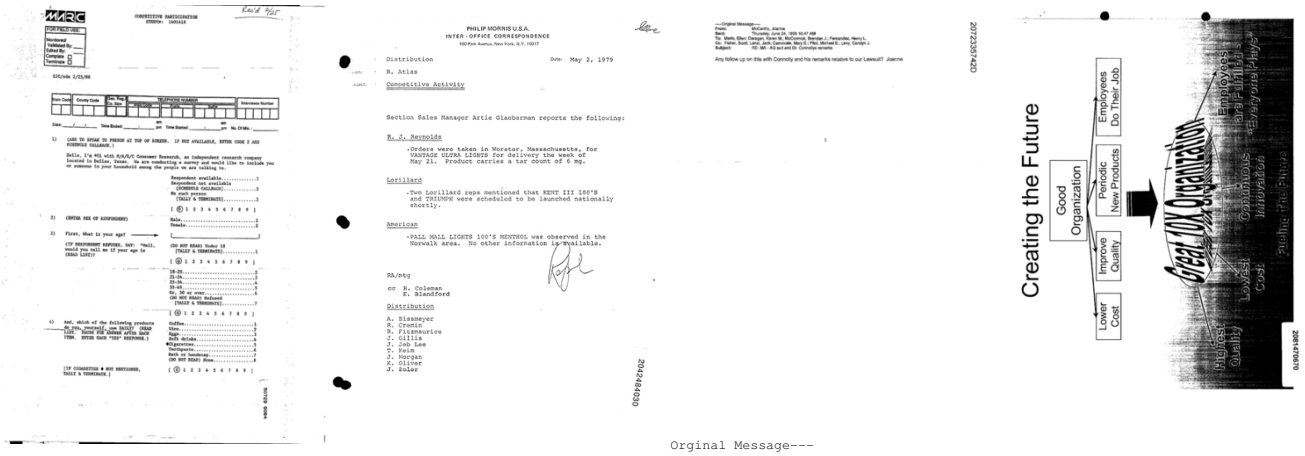


Figure 1: Multimodal classifier for hybrid text/image classification. Training is performed end-to-end on both textual and visual features.



Keck Zr COMPETITIVE
PARTICIPATION
sTupYe: 1401618 Sic/sds
2/25/88 .
= Coren aE a Dale:__/{__j__
Time Ended: 4 'pm
Time Started:. pm
No. Of Min. :, 1)
(ASK To SPEAK TO PERSON AT
TOP OF SCREEN, IF NOT
AVAILABLE, ENTER CODE 2 AY
SCHEDULE CALLBACK
...

(a) Questionnaire

PHILIP MORRIS U.S.A. koe
INTER - OFFICE
CORRESPONDENCE
100 Park Avenua, New York,
N.Y. 10017
Distribution Date:
May 2, 1979
R, Atlas
Competitive Activity

Section Sales Manager Artie
Glacberman reports the
following:
...

(b) Memo

Orginal Message---
From: McCarthy, Josnne
Sent. 'Thursday, June 24,
1990 10:47 AM
Tor Met, Eten; Daragan,
Karon M; McCormick, Brendan;
Femandez, Henry L.
(Ge: Fisher, Scot: Lon
Jock: Gamavalo, Mary ;
Pol, Michael. Ley, Ceo} J
Subject: RE: MA^{TM}-AG out
and Dr. Connolys remarks
'Any follow up on this with
Connolly and his remarks
relative to our Lawsuit?
Joanne Oztrlseezloz

(c) Email

Creating the Future
Good
Organization

Lower Improve Periodic
Employees e
Cost Quality New Products
Do Their Job

029021807

(d) Presentation

Figure 2: Document samples from the RVL-CDIP [1] dataset with corresponding text extracted by Tesseract OCR.

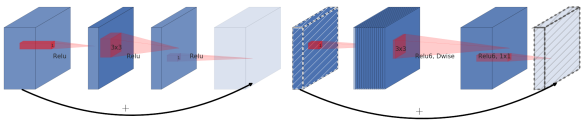
Yet, if discriminating between broad classes of documents can be achieved based on their appearance only (e.g. separating passports from banking information), fine-grained recognition often depends on the textual content of the documents. For example, different tax forms might share their layout, logos and templates while the content in itself vastly differs. Computer vision has been interested for some time in optical character recognition (OCR) to extract text from images. However, dealing with both the textual and visual contents remains an open problem. In the past years, deep learning has been established as the new state-of-the-art for image classification and natural language processing. For fine-grained document recognition, we expect the model to leverage both image and text information.

This work introduces a multimodal deep network that learns from both a document image and its textual content automatically extracted by OCR to perform its classification. We design a pragmatic pipeline for end-to-end heterogeneous feature extraction and fusion under time and cost constraints. We show that taking both the text and the document appearance into account improves both single modality baselines by several percents on two datasets from the document recognition literature. We detail some limitations of the current academic datasets and give leads for an application in an industrial setting with unclean data, such as photographed documents.

2 Related work

Analyzing digitized documents is an old task in computer vision that was boosted by the dissemination of computers in offices and then of digital cameras and smartphones in everyday life. To allow for textual search and easy indexing, the critical part of digitization is extracting text content from documents that have been scanned or photographed. Indeed, either when scanning or taking a picture of the document, its actual text is lost, although it is implicitly embedded in the pixel values of the image. Numerous optical character recognition (OCR) algorithms have been designed to transform images into strings of characters [2, 3]. Despite those efforts perfectly reading any type of document remains challenging due to the wide variety of fonts and languages. Layout analysis is a way to preprocess the data to detect text areas and find the text orientation in order to enforce a better local and global consistency [4, 5].

Document image analysis is also one of the first topic where modern deep learning has been applied. The first convolutional neural network (CNN) [6] was originally designed for classification of digits and letters. The computer vision community deployed consequent efforts to achieve image-based document classification without text, as shown by a 2007 survey [7] which focuses on document image classification without OCR results. As an example, [8] introduced SURF visual features with a bag-of-words scheme to perform document image classification and retrieval. In 2015, [1] introduced a large labeled image document dataset which sparked interest and generated several studies of deep



(a) Residual block (b) Inverted residual block

Figure 3: MobileNetV2 uses inverted residual blocks to reduce the number of channels that are forwarded in subsequent layers. Figure from [23].

CNN on this topic [9, 10, 11], inspired by the success of these networks on ImageNet and tuning data augmentation policies, transfer learning strategies and domain adaptation for document classification. In the same idea, [12] also investigated such deep architectures to classify identity documents. [13] goes even further by trying to segment the full layout of a document image into paragraphs, titles, ornaments, images etc. These models focus on extracting strong visual features from the images to classify the documents based on their layout, geometry, colors and shape.

On the other hand, text-based document classification has also long been investigated. In 1963, [14] introduced an algorithmic approach to classify scientific abstracts. More recently, [15] experimented with one-class SVM for document classification based on various text features, such as TF-IDF. [16] used Latent Dirichelet Allocation to perform topic modeling and used it as a generative approach to document classification. The recent appearance of learned word embeddings approaches such as word2vec [17] or ELMo [18] paved to way to a large body of works related to recurrent and attention mechanisms for text classification. For example, [19] proposed a bidirectional recurrent network with a hierarchical attention mechanism that learns both at the word and sentence levels to improve document classification.

Some works tried to reconcile the text-based and image-based approaches to exploit both information sources. [20] performs OCR to detect keywords in images which are then encoded as colored boxes before passing the image through a CNN. While a clever trick, this does not leverage the representation power of word embeddings. Closer to our approach, [21] goes further by generating text feature maps that are combined with visual feature maps in a fully convolutional network. However, the considered documents are synthetic and the network is trained using perfectly clean texts and images, which is unrealistic for practical uses. More similar to us, [22] learns to combine bag of words and bag of visual words features for industrial document images using a statistical model combining outputs of two single-modality classifiers. While using shallow features, they show that using both information allows for a better accuracy when the OCR is unreliable, which is often the case in an industrial setting.

In this paper, we go further in this direction and propose a new baseline with a hybrid deep model. In order to classify OCRized document images, we present a pragmatic pipeline perform visual and textual feature extraction using off-the-shelf architectures. To leverage the complementary information present in both modalities, we design an efficient end-to-end network that jointly learn from text and image while keeping computation cost at its minimum. We build on existing deep models (MobileNet and FastText) and demonstrate significant improvements using our fusion strategy on two document images dataset.

3 Learning on text and image

3.1 Visual features

There is a large literature both in general image recognition and in image document classification. Recent works have established deep convolutional neural networks as the *de facto* state of the art on many competitions in object recognition, detection and segmentation, e.g. ImageNet. Deep features, extracted by pretrained or fine-tuned deep CNNs, constitute a strong baseline for visual recognition tasks [24]. Based on this, we choose to fine-tune a CNN pretrained on ImageNet in order to extract visual features on our images, as suggested in several recent document classification publications [9, 10, 1] As we aim to perform inference on a large volume of data with time and cost constraints, we focus on a lightweight architecture with competitive classification performance, in our case the MobileNet v2 model [23].

MobileNetV2 [23] consists in a stack of bottleneck blocks. Based on the residual learning principle [25], each bottleneck block transforms a feature map first by expanding it by increasing its number of channels with a 1×1 convolutional layer with identity activation. Then, a 3×3 depthwise convolution is performed, followed by a ReLU and a final 1×1 convolution with ReLU. For efficiency issues, this block inverts the traditional residual block since the expansion is performed inside the block, whereas residual blocks compress and then reexpand the information, as illustrated in Fig. 3. The final MobileNetV2 contains 19 residual bottleneck layers. Compared to other state of the art CNNs, MobileNetV2's accuracy is on-par with VGG-16 while being significantly faster.

3.2 Textual features

Since our use case focuses on document images in which the text has not been transcribed, we need to perform an OCR step. To this end, we use the Tesseract OCR engine [3] in its 4.0 version which is based on an LSTM network. Tesseract is configured in English to use full page segmentation and the LSTM engine. In practice, this means that Tesseract will try to detect the text orientation in the image and perform the needed affine transformation and rotation if any. Tesseract also deals with the image binarization using Otsu's thresholding. This will suffice on the datasets described in Section 4.1, although we found Tesseract challenging to apply on real-world images, especially pictures which are not flat and grayscale scans.

Recent literature in NLP suggests that pretrained word embeddings offer a strong baseline which surpasses traditional shallow learning approaches. Many word embed-

dings have been designed following the initial success of *word2vec* [17], such as GloVe [26] or more recently the contextualized word embeddings from ELMo [18].
However, those word embeddings assume a good tokenization of the words, i.e. most embeddings remove digits, ignore punctuation and do not deal with out-of-vocabulary (OOV) words. Since these embeddings are learned on clean corpus (e.g. Wikipedia or novels), tokenization is fairly straightforward. OOV words are either assigned a random embedding or mapped to the closest in-vocabulary word based on the Levenshtein distance.
Unfortunately, outputs of the Tesseract OCR are noisy and not as clean as the training data from these embeddings. Even in grayscale, well-oriented documents, OCR might have trouble dealing with diacritics, exotic fonts or curved text, as illustrated by the extracts from Fig. 2. Moreover, specific user domains (e.g. banking or medieval manuscripts) might use rare words, codes, abbreviations or overall jargon that is absent from general-purpose word embeddings. Since we face many possible misspellings in the extracted text, we cannot use the previous workarounds for OOV embeddings since it would inject a lot of non-discriminant features in our text representation In average, on the Tobacco3482 corpus, a document processed by Tesseract OCR contains 136 words with 4 characters or more. Of those, only 118 are in the 20k GloVe embeddings [26] and only 114 are in Enchant's spellchecker US English dictionary. Overall, approximately 26% of the corpus is absent from the US English dictionary and 23% from the GloVe embeddings. The document distribution with respect to the proportion of out-of-vocabulary words is shown in Fig. 4a. Although most of the documents are concentrated around 10% of OOVs, there is a significant long tail including several dozens of documents that contain only words outside of the English language.
Therefore, we turn to character-based word embeddings that are able to deal with OOV words by assigning them plausible word vectors that preserve both a semantic and a spelling similarity. One possibility was to use the mimicking networks from [27] that learn to infer word embeddings such as GloVe, but based only on subword information. More complex embeddings such as FastText [28, 29] and ELMo [18], which produce vectors using respectively n-grams and subword information, can also address this problem. Finally, the Magnitude library [30] uses two alternative strategies to deal with OOV words:

- Assigning a *deterministic* random vector. These vectors do not capture semantic sense, however similar words based on the Levenshtein-Damerau distance will have similar vectors. Misspellings will therefore not be close to the original word, but similar lingo words will be close.

- Using character n-grams inspired by [28] and interpolation with in-vocabulary words, Magnitude can generate vectors for OOV words which are sensible based on existing learned embedding.

Preliminary data exploration shows that subword-aware embeddings perform better at preserving similarity despite misspellings, as illustrated in Fig. 4b. We therefore focus our interest on the FastText embedding, which is faster than ELMo since the latter requires passing the context through a bidirectionnal LSTM during inference.
Finally, it is necessary to convert those word embeddings into a document embedding. We consider two approaches:

- The simple baseline for sentence embedding suggested in [31], which consists in a weighted average of word embeddings altered by PCA.

- Using variable-length document embeddings consisting in a sequence of word embeddings.

The first approach is suitable as generic feature while the second requires a statistical model able to deal with sequences, such as recurrent or convolutional neural networks. For both methods, we use the SpaCy small English model [32] to perform the tokenization and punctuation removal. Individual word embeddings are then inferred using FastText [28].

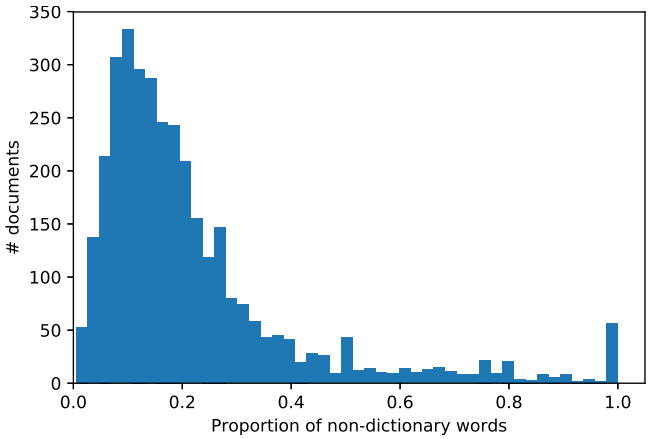
3.3 Multimodal features

Once text and image features have been extracted, we feed them to a multi-layer perceptron following [33]. To do so, we need to combine both feature vectors into one. Two approaches can be envisioned:

- Adaptive averaging of both feature vectors. This aligns both feature spaces so that scalars at the same index become compatible by summation, i.e. that each dimension of the vectors have a similar semantic meaning.

- Concatenating both vectors. This does not imply that both feature spaces can be aligned and delegates to the fusion MLP the task of combining the two domains.

Both fusion strategies are differentiable, therefore the whole network can be trained in an end-to-end fashion. Moreover, the model is modular and each feature extractor can be swapped for another model, e.g. MobileNet can be exchanged with any other popular CNN and FastText could be replaced by subword-level NLP models, even differentiable ones that could allow fine-tuning the embeddings. In this work, we try to keep things simple and build on robust base networks in order to clearly understand how the data fusion impacts model performance. Preliminary experiments showed that the summation fusion significantly underperformed compared to pure image baseline. We suggest that this is provoked by the impossibility of aligning the text and image feature spaces without breaking their discriminating power, resulting in suboptimal space. Therefore, we move on with the concatenation strategy for the rest of this paper. The complete pipeline is illustrated in Fig. 1.



(a) Document distribution w.r.t of non-dictionary words % in the Tobacco3482-Tesseract corpus.

Word pair	Similarities		
`specifically`	GloVe	ELMo	FastText
`Specificalily`	0.71	0.68	0.96
`filter`	GloVe	ELMo	FastText
`fiilter`	0.91	0.73	0.96
`alcohol`	GloVe	ELMo	FastText
`Aleohol`	0.40	0.69	0.88
`Largely`	GloVe	ELMo	FastText
`Largly`	0.25	0.81	0.98

(b) Word embeddings similarity for misspelled words.

Figure 4: Tesseract OCR outputs noisy text that does not entirely overlap with the assumptions usually held when training word embeddings for NLP.

4 Experimental setup

4.1 Datasets

Tobacco3482. The Tobacco3482 dataset [8] contains 3482 black and white documents, a subset from the Truth Tobacco Industry Documents[1] archives of legal proceedings against large American tobacco companies. There are annotations for 10 classes of documents (e.g. email, letter, memo...). Following common practices, we perform k-fold cross-validation using 800 documents for training and the rest for testing. Results are averaged over 3 runs.

RVL-CDIP. The RVL-CDIP dataset [1] is comprised of 400 000 grayscale digitized documents from the Truth Tobacco Industry Documents. There are annotations for 16 classes of documents (e.g. email, letter, invoice, scientific report...), each containing 25 000 samples. We use the standard train/val/test split from [1] with 320 000 documents for training, 40 000 for validation and 40 000 for testing.

4.2 Models

This subsection describes the implementation details of our deep networks. All models are implemented in TensorFlow 1.12 using the Keras API and trained using a NVIDIA Titan X GPU. Hyperparameters were manually selected on a subset of Tobacco3482 and fixed for all experiments.

Text baseline. Seeing that our representation of textual data can be either a document embedding or a sequence of word embeddings, we compare two models for our text baseline.

The first model is an improved Multi-Layer Perceptron (MLP) with ReLU activations, Dropout and Batch Normalization (BN) after each layer. The network has a fixed width of 2048 neurons for all layers except the last one, which produces a 128 feature vector, classified by a softmax layer. Weights are randomly initialized using He's initialization [34]. The averaged document embedding [31] is used as an input for this classifier.

The second model is a one-dimensional convolutional neural network designed inspired by previous work for sentence classification [35]. The CNN is 4-layers deep and interlaces 1D convolutions with a window of size 12 with maxpooling with a stride of 2. Each layer consists in 512 channels with ReLU activation. The final feature map is processed by a max-pooling-through-time layer that extracts maximal features on the sequence on top of which we apply Dropout for regularization. A fully connected layer then maps the features to the softmax classifier. The input word sequence is zero-padded up to 500 words for documents with less 500 words.

We experiment on the Tobacco3482 dataset in order to evaluate which text model to choose. Results are reported in Table 1a. Without surprise, the CNN 1D outperforms significantly the MLP classifier. The pattern recognition abilities of the convolutional network makes it possible to interpret the word sequences by leveraging contextual information. Since only some part of the text might be relevant, averaging over all word embeddings dilute the discriminating information. Moreover, noisy embeddings due to garbage output from Tesseract (e.g. incoherent strings where OCR has failed) are included in the final document embedding. However, when dealing with word sequences, convolutional layers and temporal max-pooling help extracting only the relevant information. Therefore, we choose to include the 1D CNN as the text component in our multimodal architecture.

This model is denoted **TEXT** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 100 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40.

Image baseline. We investigate as our base CNN the lightweight MobileNetV2 [23] which focuses on comput-

[1] `https://www.industrydocuments.ucsf.edu/tobacco/`

(a) Preliminary experiments on Tobacco3482 for the text baseline.

Model	OA	F_1
MLP (document)	70.8%	0.69
CNN 1D (word sequence)	73.9%	0.71

OA = overall accuracy, F_1 = class-balanced F_1 score.

(b) Preliminary experiments on Tobacco3482 for the image baseline.

Model	OA	F_1
MobileNetV2	84.5%	0.82
MobileNetV2 (w/ DA)	83.9%	0.82

OA = overall accuracy, F_1 = class-balanced F_1 score, DA = data augmentation.

Table 1: Preliminary tuning of the single-modality baselines on Tobacco3482.

ing efficiency, albeit at the cost of a slightly lower top-1 accuracy on ImageNet compared to other state of the art CNN. We train the CNN on grayscale document images resized at 384×384. Although this warps the aspect ratio, [9] reports better accuracy than when using padding at the same resolution. As the model is designed for RGB images, the grayscale channel is duplicated three times. This allows us to initialize the network by loading its pretrained weights on ImageNet, which accelerates convergence and slightly improves accuracy through transfer learning.

This model is denoted **IMAGE** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 200 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40.

As reported in Table 1b, preliminary experiments on the Tobacco3482 with random JPEG artifacts, saturation and contrast alterations did not significantly alter the classifier's accuracy compared to no augmentation. This is explained by the low variability between the grayscale document images. All images are grayscale with dark text on white background with horizontal text lines, therefore color and geometric augmentation are not necessary. However, [9] report some success using shear transform, which we did not consider in this work. It is worth noting that compared with previous literature on the RVL-CDIP dataset, e.g. [9, 10, 1], we do not average predictions over multiple crops at inference time for speed concerns.

Fusion. For our multimodal network, we consider the same model as our baselines except that the final layers are cut-off. For the **TEXT** model, the last layer produces an output vector of dimension 128 instead of the number of classes. For the **IMAGE** model, we aggregate the last convolutional features using global average pooling on each channel, which produces a feature vector of dimension 1280. We then map this feature vector using a fully connected layer to a representation space of dimension 128.

This model is denoted **FUSION** in the rest of the paper. It is optimized using Stochastic Gradient Descent with momentum for 200 epochs, with a learning rate of 0.01, a momentum of 0.9 and a batch size of 40.

5 Discussion

5.1 Performances

Model performances scores on Tobacco3482 and RVL-CDIP are reported in Tables 2 and 3.

Behaviour of all models is consistent both on the smaller dataset and on the very large one. In both cases, the **TEXT** baseline is significantly underperforming the **IMAGE** one. Indeed, as could be seen in Fig. 2, Tesseract OCR outputs noisy text. This includes words that have been misspelled – which are correctly dealt with by the FastText embeddings – and new words that are hallucinated due to poor binarization or salt-and-pepper noise in the image. Moreover, layout and visual information tends to be more informative based on how the classes were defined: scientific papers, news and emails follow similar templates while advertisements present specific graphics. However, in both cases, this simple document embedding is enough to classify more than 70% of the documents, despite its roughness.

Using the **IMAGE** model only, we reach accuracies competitive with the state of the art. MobileNetV2 alone does on-par is with the holistic CNN ensemble from [1] and is competitive with fine-tuned GoogLeNet and ResNet-50 [10] (90.97%).

On both datasets, the fusion scheme is able to improve the overall accuracy by $\simeq 1.5\%$ which demonstrates the relevance of our approach. While the document embedding we chose is simple, it appears to be at least partially robust to OCR noise and to preserve enough information about the document content to boosts CNN accuracy on document image classification even further. We also report the results from an oracle, which corresponds to the perfect fusion of the **TEXT** and **IMAGE** baselines, i.e. a model that would combine the predictions from both single-modality networks and always choose the right one. The oracle corresponds to the theoretical maximal accuracy boost that we could expect from the **FUSION** model. On Tobacco3482, the oracle corresponds to a 7.6% absolute improvement (9% relative). In our case, the **FUSION** model improves the best single-source baseline by an absolute 3.3% (4% relative), which is significant although still leaves the door open to further improvements. More importantly, the gains are consistent on all classes of interest, almost never underperforming one of the two base networks on any class. This confirm the proposed approach as the two sources, image and text, give complementary information to classify a document.

5.2 Processing time

Although some applications of document image recognition can be performed offline, most of the time users upload a document and expect near real-time feedback. User experience engineering [36] indicates than less than 1s is the maximum latency the user can suffer before the interface feels sluggish, and 10s is the maximum delay before they

Table 2: Overall accuracy on the RVL-CDIP dataset.

Model	IMAGE	TEXT	FUSION	CNNs [1]	VGG-16 [10]	AlexNet+SPP [9]
OA	89.1%	74.6%	90.6%	89.8%	90.97%	90.94%

OA = Overall Accuray.

Table 3: Overall accuracy and F_1 scores on the Tobacco3482 datasets.

Model	OA	F_1	Adv.	Email	Form	Letter	Memo	News	Notes	Report	Res.	Sci.
CNNs [1]	79.9	–						–				
TEXT	73.8	0.71	0.60	0.96	0.76	0.71	0.79	0.67	0.62	0.43	**0.97**	0.57
IMAGE	84.5	0.82	**0.94**	0.96	0.85	0.83	0.90	0.89	0.83	0.61	0.80	0.62
FUSION	**87.8**	**0.86**	0.93	**0.98**	**0.88**	**0.86**	**0.90**	**0.90**	**0.85**	**0.71**	0.96	**0.68**
Oracle	92.1	0.91	0.94	0.99	0.94	0.92	0.93	0.93	0.89	0.81	0.97	0.79

Adv. = Advertisement, Res. = Resume, Sci. = Scientific.

start loosing their attention. On the RVL-CDIP dataset, Tesseract OCR processes a document image in $\simeq 910\text{ms}$ in average on an Intel Core i7-8550U CPU using four threads, including loading the image from disk. This means that every additionnal latency induced by the network inference time is critical since it will negatively affect the user experience.

On the same CPU, the full inference using the **FUSION** model takes $\simeq 360\text{ms}$ including loading, resizing and normalizing the image. The complete process including Tesseract OCR therefore takes less than $\simeq 1300\text{ms}$ which is acceptable in a system requiring user input. Of those, 130ms are spent in the 1D CNN (including reading the file and performing FastText inference) and 230ms in MobileNetV2 (including image preprocessing). The overhead added by the final fusion layer is negligible. We stress that this is using a standard TensorFlow without any CPU-specific compilation flags, which could speed up the inference further. On a NVIDIA Titan X GPU, the **FUSION** network runs in 110ms (50ms for **TEXT**, 60ms for MobileNetV2), which brings the total just above the 1s recommendation. In our case, using compute-efficient architectures allow us to avoid running on an expensive and power-hungry GPU.

As a comparison basis, other architecture choices that we dismissed earlier would have resulted in poorer performance and the network would not be usable in a near real-time user application. For example, the Xception network [37] takes 630ms to run during inference with the same parameters and hardware. For the text model, an LSTM-based RNN with a similar depth takes many seconds to run.

Note that, although this does not reduced the perceived delay for one user, the global throughput of the system can be improved by batching the images. Two Tesseract processes can leverage the full eight cores from an Intel Core i7-8550U CPU. In this setting, processing an image takes $\simeq 660\text{ms}$ in average. Thanks to the batch efficiency of neural networks, the average processing time becomes $\leq 750\text{ms}$ on GPU and $\leq 1000\text{ms}$ on CPU. This is particularly helpful when users have several documents to upload that can be processed concurrently.

5.3 Limitations

One of the main limitation of this work stems from the public document image datasets available. Indeed, in a real-world application, document images can be grayscale, RGB, scanned images and photographs with various rotations, brightness, contrast and hue values. The Tobacco documents are all oriented in the right way, which makes it easier for Tesseract to perform OCR. Moreover, documents have been scanned by professionals who tried to maximize their legibility while user-generated often presents poor quality.

While it was not required here, data augmentation is definitely required for practical applications to encompass the large variety of environmental conditions in which documents are digitized. This is especially true for rotations, since it is often not possible to ensure that users will capture the document with the right orientation and Tesseract does not always correctly detects it. For industrial-grade applications dealing with user-generated content, such a data augmentation is necessary to alleviate overfitting and reduce the gap between train and actual data. Preprocessing page segmentation and layout analysis tools, such as dhSegment [13] can also bring significant improvements by renormalizing image orientation and cropping the document before sending it to the classifier.

Moreover, as we have seen, the post-OCR word embeddings include lots of noisy or completely wrong words that generate OOV errors. In practical applications, we found beneficial to perform a semantic tokenization and named entity recognition using SpaCy. This allows us to perform a partial spellchecking, e.g. using symspell [2] to correct words that have been misread by Tesseract, without affecting proper nouns or domain-specific abbreviations and codes. If this can deal frequent mispellings of words, it might also suppress out-of-vocabulary words such as alphanumeric codes. Therefore, learning domain specific,

[2] https://github.com/wolfgarbe/SymSpell

character-based or robust-to-OCR embeddings [38] is an interesting lead for future research, as the current interest in the ICDAR2019 competition on Post-OCR Text Correction shows[3].

6 Conclusion

In this work, we tackled the problem of document classification using both image and text contents. Based only on an image of a digitized document, we try to perform a fine-grained classification using visual and textual features. To do so, we first used Tesseract OCR to extract the text from the image. We then compute character-based word embeddings using FastText on the noisy Tesseract output and generate a document embedding which represents our text features. Their counterpart visual features are learned using MobileNetv2, a standard CNN from the state of the art. Using those pragmatic approaches, we introduce an end-to-end learnable multimodal deep network that jointly learns text and image features and perform the final classification based on a fused heterogeneous representation of the document. We validated our approach on the Tobacco3482 and RVL-CDIP datasets showing consistent gains both on small and large datasets. This shows that there is a significant interest into hybrid image/text approach even when clean text is not available for document image classification and we aim to further investigate this topic in the future.

References

[1] A. W. Harley *et al.*, "Evaluation of deep convolutional nets for document image classification and retrieval," in *ICDAR*, Aug. 2015.

[2] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document Analysis System," *IBM J. Res. Dev.*, vol. 26, Nov. 1982.

[3] A. Kay, "Tesseract: An Open-source Optical Character Recognition Engine," *Linux J.*, vol. 2007, July 2007.

[4] D. X. Le *et al.*, "Classification of binary document images into textual or nontextual data blocks using neural network models," *Mach. Vis. Appl.*, vol. 8, Sept. 1995.

[5] S. Imade *et al.*, "Segmentation and classification for mixed text/image documents using neural network," in *ICDAR*, Oct. 1993.

[6] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, Nov. 1998.

[7] N. Chen and D. Blostein, "A survey of document image classification: Problem statement, classifier architecture and performance evaluation," *Int. J. Doc. Anal. Recogn.*, vol. 10, June 2007.

[8] J. Kumar *et al.*, "Structural similarity for document image classification and retrieval," *Pattern Recognit. Lett.*, vol. 43, July 2014.

[9] C. Tensmeyer and T. Martinez, "Analysis of Convolutional Neural Networks for Document Image Classification," in *ICDAR*, vol. 01, Nov. 2017.

[3] https://sites.google.com/view/icdar2019-postcorrectionocr

[10] M. Z. Afzal *et al.*, "Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification," in *ICDAR*, vol. 01, Nov. 2017.

[11] A. Das *et al.*, "Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks," in *ICPR*, Aug. 2018.

[12] R. Sicre *et al.*, "Identity Documents Classification as an Image Classification Problem," in *ICIAP*, 2017.

[13] S. Ares Oliveira *et al.*, "dhSegment : A generic deep-learning approach for document segmentation," Aug. 2018.

[14] H. Borko and M. Bernick, "Automatic Document Classification," *J. ACM*, vol. 10, Apr. 1963.

[15] L. M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *J. Mach. Learn. Res*, vol. 2, Dec. 2001.

[16] T. N. Rubin *et al.*, "Statistical topic models for multi-label document classification," *Mach. Learn.*, vol. 88, July 2012.

[17] T. Mikolov *et al.*, "Efficient Estimation of Word Representations in Vector Space," in *ICLR*, Jan. 2013.

[18] M. Peters *et al.*, "Deep Contextualized Word Representations," in *NAACL*, 2018.

[19] Z. Yang *et al.*, "Hierarchical Attention Networks for Document Classification," in *NAACL*, 2016.

[20] L. Noce *et al.*, "Embedded Textual Content for Document Image Classification with Convolutional Neural Networks," in *ACM Symposium on Document Engineering*, 2016.

[21] X. Yang *et al.*, "Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks," in *CVPR*, July 2017.

[22] O. Augereau *et al.*, "Improving Classification of an Industrial Document Image Database by Combining Visual and Textual Features," in *IAPR Workshop*, Apr. 2014.

[23] M. Sandler *et al.*, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *CVPR*, June 2018.

[24] A. S. o. Razavian, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *CVPRW*, June 2014.

[25] K. He *et al.*, "Deep Residual Learning for Image Recognition," in *CVPR*, June 2016.

[26] J. Pennington *et al.*, "Glove: Global Vectors for Word Representation," in *EMNLP*, 2014.

[27] Y. Pinter *et al.*, "Mimicking Word Embeddings using Subword RNNs," in *EMNLP*, 2017.

[28] P. Bojanowski *et al.*, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, 2017.

[29] A. Joulin *et al.*, "Bag of Tricks for Efficient Text Classification," in *EACL*, 2017.

[30] A. Patel *et al.*, "Magnitude: A Fast, Efficient Universal Vector Embedding Utility Package," in *EMNLP*, Nov. 2018.

[31] S. Arora *et al.*, "A Simple but Tough-to-Beat Baseline for Sentence Embeddings," in *ICLR*, Nov. 2016.

[32] M. Honnibal and Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[33] A. Eitel *et al.*, "Multimodal deep learning for robust RGB-D object recognition," in *IROS*, Sept. 2015.

[34] K. He *et al.*, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *CVPR*, Dec. 2015.

[35] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *EMNLP*, Oct. 2014.

[36] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[37] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *CVPR*, July 2017.

[38] V. Malykh *et al.*, "Robust Word Vectors: Context-Informed Embeddings for Noisy Texts," in *EMNLP W-NUT*, Nov. 2018.