

# Image and Text fusion for UPMC Food-101 using BERT and CNNs

Ignazio Gallo, Gianmarco Ria, Nicola Landro, and Riccardo La Grassa

Department of Theoretical and Applied Science, University of Insubria, Varese, Italy  
{ignazio.gallo, gria, nlandro, rlagrassa}@uninsubria.it

**Abstract**—The modern digital world is becoming more and more multimodal. Looking on the internet, images are often associated with the text, so classification problems with these two modalities are very common. In this paper, we examine multimodal classification using textual information and visual representations of the same concept. We investigate two main basic methods to perform multimodal fusion and adapt them with stacking techniques to better handle this type of problem. Here, we use UPMC Food-101, which is a difficult and noisy multimodal dataset that well represents this category of multimodal problems. Our results show that the proposed early fusion technique combined with a stacking-based approach exceeds the state of the art on the dataset used.

## I. INTRODUCTION

Classification is one of the most important tasks in Data Mining and has become extremely popular above the Data Science community. It consists of a function that assigns items in a collection to target categories, usually called *classes*. The final goal of classification is to accurately predict the target class for each case in the data. Classification includes Images classification, a supervised process of labelling images according to predefined categories. The classification model is normally fed a set of images within a specific category and, based on this set, it can learn which class the images belong to. A common deep learning method for image classification is to train an Artificial Neural Network (ANN) to process input images and predict a class for each image. The training process can be extremely long and expensive, however, Convolutional Neural Networks (CNN) excel at this kind of task. Different techniques and approaches have been proposed to increase the classification performances; one of them is the Multimodal Classification. We are well aware that data in real-world scenarios usually come as different modalities: images are usually associated with tags and text explanations, texts contain images to more clearly express their meaning. On e-commerce websites, for each item on sale, a user can select a product based on a text and image that shows characteristics, colours and other features of the product [1] and in general in the literature exists different approaches to multimodal classification [2] [3]. Different modalities are characterized by very different statistical properties (e.g. images represented as pixel intensities or outputs of feature extractors and texts

represented as discrete word count vectors). It is crucial to discover the relationship between different modalities of the considered data. The multimodal learning model is capable to fill missing modality, given the observed ones. In the case of ambiguous images, it is possible to disambiguate the wrong classifications and improve the results by combining text and images. Different techniques can be applied to create a multimodal classifier and the outcomes usually depend on the kind and quality of data taken into consideration.

In this work we propose a novel technique of multimodal classification, using for our experiments the dataset introduced in [4] at the *Pierre and Marie Curie* University of Paris, the **UPMC Food-101**.

First, we develop and test different techniques to classify the images belonging to the dataset, to obtain the highest level of accuracy. Secondly, we focus our attention on the textual part of the dataset to create an algorithm to classify the text related to each image to the correct class. Finally, we propose a technique to combine two classifiers into a new multimodal classifier that reaches a higher level of accuracy compared to the single employed classifiers.

## II. RELATED WORK

The classification model we develop for images is based on Convolutional Neural Networks and, in particular, on Inception. Since 2014, deep Convolutional Neural Networks started to become mainstream and the use of deeper and wider architectures allowed to improve the performances and results on many different tasks. VGGNet [5] and GoogLeNet [6], for instance, yielded high performances in the 2014ILSVRC classification challenge. However, the computational cost of GoogLeNet [6] is much lower compared to the VGGNet [5] and this makes the network able to perform well even under strict memory constraints and computational budget. Still, the complexity of the Inception architecture employed in GoogLeNet makes it more difficult to make changes to the network. We choose to base our classification model on the Inception V3 architecture proposed by Szegedy et al. [7], seen that it reaches state-of-the-art results on the ILSVRC 2012 [6] classification benchmark, lowering the error score compared to other known networks such as GoogLeNet [6], VGG [5], BN-Inception [8], and PReLU [9].

The model that we develop for the classification of images-related texts is based on Bidirectional Encoder Representations

from Transformers (BERT), defined in the paper proposed by Devlin et. al. [10] at Google AI Language. BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers [10]. New state of the art results has been obtained by BERT on important languages processing tasks such as GLUE (7.7% point absolute improvement), MultiNLI (4.6% absolute improvement) SQuAD v1.1 and v2.0 (1.5% and 5.1% points improvement respectively). This is the reason why we choose to develop a classification algorithm that includes the new techniques introduced with BERT [10], including input representations and tokenization.

In literature exists some multimodal approach for the UPMC Food-101 dataset. Wang et al. [4], present deep experiments of recipe recognition on the dataset using visual information, textual information and the fusion of both. For the images classification, they employ CNN and the pre-trained model imagenet-vgg-verydeep19 [11]. We instead develop a custom model based on Inception V3 [7]. As regarding textual features, they propose the usage of TF-IDF (Term Frequency-Inverse Document Frequency) technique, whereas we make use of the pre-processing available with BERT [10]. Finally, they merge very deep features and TF-IDF classification scores by late fusion, where the fusion score  $s_f$  is a linear combination of the scores provided by both image and text classification systems ( $s_f = \alpha s_i + (1 - \alpha) s_t$ ). In our approach we first extract and merge features from images and text, then we *stack* a classifier.

Kiela et al. [12] investigate various methods for performing multi-modal fusion, analyzing their trade-offs in terms of classification accuracy and computational efficiency. They use FastText [13] and ResNet-156 [14] for the two modalities: text and image classification. Instead we use BERT [10] for text classification and InceptionV3 [7] for image classification. The last difference is the fusion technique: they obtain the best result by using bilinear-gated fusion instead of in our best results we use early fusion with a stacking approach [15].

Narayana et al. [16] propose a novel method called HUSE, "Hierarchical Universal Semantic Embeddings", that projects images and text into a shared latent space by using a shared classification layer for image and text modalities. They obtain embeddings both from images and texts; in particular, they use BERT embeddings to gain a representation of the text, extracting salient tokens from the complete web page, and they extract pre-trained Graph-Regularized Image Semantic Embeddings (Graph-RISE) from each image to obtain a representation of the images. Instead, we do not make use of embeddings for images, while for the texts we make use of the web-pages titles instead of extracting tokens from the full page.

### III. PROPOSED APPROACH

The focus of this paper is to create a multimodal classifier building two different models. Once developed the two models, we apply the *stacking* technique [15], where the simple classifiers are stacked on top of each other so as to learn

complex functions or other classifiers. The model we define for images classification is a Convolutional Neural Network based on the Inception V3 architecture. This choice comes from the willingness to reach a compromise between performances (in terms of number of parameters) and achievable accuracy. Looking at the Keras Applications page [17] it is possible to observe that Inception V3 is able to reach good accuracy results on ImageNet dataset, while keeping the number of parameters relatively low. In the interest of successfully applying the Inception V3 architecture to our task, we make use of the transfer learning. The Inception V3 model [7] has been trained on the huge ImageNet dataset, therefore the learned weights can be used as the starting point for the training process of the the new network for the Food-101 classification. We remove the two last layers from the original implementation: the GlobalAveragePooling2D and the classification layer (fully connected). Following the inception model, we add an average pooling layer having a filter size of  $8 \times 8$ . After the pooling layer, we insert a dropout layer, with a probability of 40%. Finally, there is the classification layer: a dense layer (fully-connected) having 101 neurons, since we are dealing with 101 food classes. The activation function is the *softmax*, which assigns decimal probabilities to each possible class. The final

TABLE I: Custom model defined for the classification of UPMC Food-101 [4] images

Model for UPMC Food-101 Images Classification
Input Image - shape (299 x 299 x 3)
Inception V3 (without the last two layers)
Average Pooling 2D (8 x 8)
Dropout - probability 40%
Flatten
Dense <i>Softmax</i> - 101 units

image classification model used in this paper is available in Table I.

The next step is the search for a performing technique for the classification of UPMC Food-101 [4] images-related text documents. To obtain the best possible result, we decide to explore and apply new state-of-the-art techniques, that is the BERT approach [10]. For training time and performance reasons, we decide to use the *BASE* version of the BERT model having the following parameters: English language uncased, 12 hidden layers (L), 768 hidden size (H), 12 self-attention heads (A), 30522 words dictionary (vocab\_size), 110 millions parameters in total. The classification model will clearly have a BERT layer as first layer, and this requires a pre-processing of input texts to make them suitable for the model. We decide to combine the power of BERT with the capabilities of Recurrent Neural Networks (RNN). Long Short Term Memory networks (LSTMs) are a special kind of RNNs, capable of learning long-term dependencies and explicitly designed to avoid the long-term dependency problem, since remembering information for long periods of time is their default behavior. The approach for text classification is designed as shown in Table II. Looking at Table II we can see that the output of the BERT layer is passed to an LSTM layer having 128 units. We found that 128

TABLE II: Custom model defined for the classification of UPMC Food-101 [4] texts

Model for for UPMC Food-101 Texts Classification
Inputs (word ids, masks, segment ids)
BERT Model
LSTM - 128 units
Dropout - probability 50%
Dense - 256 units
Dropout - probability 50%
Dense <i>Softmax</i> - 101 units

is the best choice in view that (1) it allows to reach the highest level of accuracy while (2) keeping overfitting under control and (3) reducing the complexity and training time. Before the last classification layer, we add a fully-connected layer having 256 units and two dropout layers with 50% probability. Once having developed a model for each data source, the most important question to answer is "what is the best strategy to fuse our multiple modalities?". In literature, the fusion of different modalities is generally performed at two levels: *feature level* or **early fusion**, and *decision level* or **late fusion**.

In late fusion approaches, each modality model analyze individual features  $F_1, \dots, F_n$  and outputs local decisions  $D_1, \dots, D_n$ . The local decisions are then combined using a Decision Fusion unit (DF) to create a fused *decision vector*. This vector is subsequently analyzed by a new Analysis Unit, in order to obtain the final decision D about the task.

In our work we have two Analysis Units (AU): one for each modality. The AU for the images is the model based on Inception V3, whereas the AU for the text is the BERT + LSTM model. Having these two models trained on images and texts, a combiner algorithm is needed for making the final prediction using their decisions as inputs. During our experiments we try and evaluate different *Stacking* methods [1], as reported in Table VI. The new multimodal model we implement using the late-fusion technique is shown in Figure 1 and a summary of the model is available in Table III.

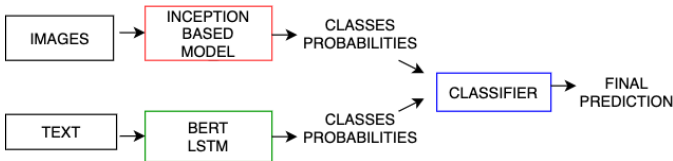


Fig. 1: Scheme of the implemented **late fusion** model with a stacking approach.

In the feature level multimodal fusion, or early fusion approach, the features are extracted from input data by the single Analysis Units. Later, features coming from different modalities are combined by a special Feature Fusion (FF) unit. In a feature level multimodal analysis task, the extracted features are first fused using a Feature Fusion unit and then the combined feature vector is passed to an Analysis Unit (AU) for the actual analysis.

So, we use the Inception-based model for images by removing

TABLE III: Late fusion model architecture.

Input Image	Input Text
Inception V3	BERT Model
Average Pooling 2D (8 x 8)	LSTM - 128 units
Dropout - probability 40%	Dropout - probability 50%
Flatten	Dense - 256 units
	Dropout - probability 50%
Dense <i>Softmax</i> - 101 units	Dense <i>Softmax</i> - 101 units
Concatenate	
Dense - 256 units	
Dropout - probability 20%	
Dense - 128 units	
Dropout - probability 20%	
Dense <i>Softmax</i> - 101 units	

the *softmax* activation function from the last Dense layer and by setting the number of units to 128 (instead of 101, that is the number of classes).

As regarding the BERT + LSTM model for text classification, we modify the original implementation by removing the last four layers, in particular: the Dropout, the Dense with 256 units, the second Dropout and the final classification layer (Dense with 101 units). The LSTM is the remaining last layer in the model, that outputs the features extracted from text by the BERT and LSTM architectures without making predictions about the class.

We choose to use a concatenation layer to merge the features extracted from the images by the Inception model to the features extracted from the text by the BERT + LSTM model, creating a vector of fused features.

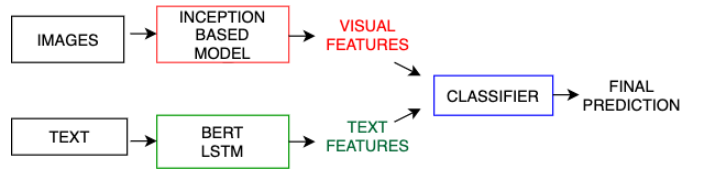


Fig. 2: Scheme of the implemented **early fusion** model with a stacking approach.

Table IV shows the final implementation. The Inception model's last layer is a Dense with 128 units and, for the BERT model, its last layer is a LSTM with 128 units. These outputs are passed to the concatenation layer. Following the concatenation, we add a Dense layer with 256 units and *ReLU* activation function. Finally we can see the classification layer, that is a Dense with 101 units and *softmax* activation. This last layer will actually output the final decision D about the predicted class of input image and text.

#### IV. DATASET

Our experiments are based on the very large and challenging UPMC Food-101 multimodal dataset [4]: it is a classification dataset that contains images and texts. UPMC Food-101 shares the same categories with one of the largest, but more simple, public food image dataset: the *ETHZ Food-101* [18]. In the UPMC Food-101 all the images are collected in an uncontrolled environment, making it the noisiest Food-101 dataset.

TABLE IV: Proposed model architecture.

Input Image	Input Text
Inception V3	
Average Pooling 2D (8 x 8)	
Dropout - probability 40%	
Flatten	BERT Model
Dense - 128 units	LSTM - 128 units
Concatenate	
Dense - 256 units	
Dense <i>Softmax</i> - 101 units	



(a) Real Sashimi



(b) Multiple plates



(c) Random image



(d) Wrong class

Fig. 3: Four images taken from the sashimi class of UPMC Food-101. In (a) a real image of sashimi; the image in (b) contains more dishes so it's hard to understand if it's sashimi; In the sub-figure (c) an example of the sashimi class containing a random image; while in (c) we have a wrong example containing the image of another class of the same dataset.

Each category is estimated to contain about 5% irrelevant images because no human intervention was applied during the dataset acquisition. Figure 3 contains four images taken from the 'sashimi' class. As we can see, Figure 3a is a correct sashimi plate, while the other figures are noise. In particular, Figure 3b contains multiple plates, Figure 3c is a random image that has nothing to do with sashimi and, lastly, Figure 3d is an image belonging to another class.

In the Table V we can see a comparison between the UPMC Food-101 and the ETHZ. The UPMC Food-101 dataset is composed of a training set containing 67,988 samples and a test set containing 22,716 samples, for a total of 90,704 images and text documents belonging to 101 classes. In particular, for the text part, we use only the title of text document.

TABLE V: Comparison between ETHZ [18] and UPMC [4] Food-101 dataset

UPMC and ETHZ Food-101 comparison			
Dataset	Number of Images per class	Data Type	Source Environment
ETHZ	1000	Images	Controlled
UPMC	790 - 956	Images + Text	Not Controlled

## V. EXPERIMENTS

We train all the described models on *Google Colaboratory* using an NVIDIA Tesla P100 GPU equipped with 16 GB of memory. First, we train the InceptionV3-based model defined for the classification of images only. Through the experiments we discover that the best optimizer for this task is *SGD* (*Stochastic Gradient Descent*). As regarding the learning rate, we start the training with a higher value and then we lower it as the number of epochs increases (we start from a value of 0.01, then lower it to 0.001 and finally to 0.0005). The loss function used is the *cross-entropy*.

Figure 4 shows the progress of test classification accuracy for all the implemented models: InceptionV3-based for images (blue), BERT LSTM for texts (red), Late Fusion (green) and Early Fusion (purple). On *x-axis* we have the epoch number while on *y-axis* we have the accuracy score. Our images model reaches 71.67% accuracy on test set, with no overfitting observed.

Secondly, we train the BERT + LSTM model (in Table II), in this case using *Adam* optimizer starting with a learning rate equal to 0.001 and decreasing it using the *Reduce Learning Rate on Plateau* method (if no improvement is seen for a patience number of epochs, the learning rate is reduced by a specified factor).

Looking at Figure 4 we can see that this model is able to reach 84.41% of test accuracy. After the 26<sup>th</sup> epoch, no improvement in test accuracy is observed, therefore the training process is stopped. Finally, we train and evaluate the multimodal

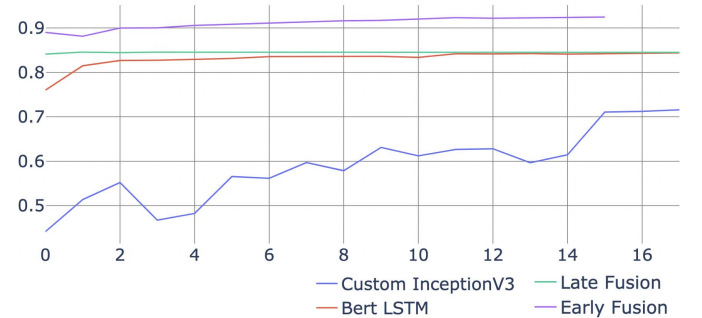


Fig. 4: Progress of classification accuracy on UPMC Food-101 test set

classifiers, starting with the late-fusion approach (Figure 1). We train the model using *SGD* optimizer, lowering the learning rate from 0.01 to 0.001 and finally to 0.0001 during the epochs. From Figure 4 it is possible to notice that test accuracy starts with a relatively high value from the 1<sup>st</sup> epoch, and then shows only minimal improvements during the training. The best result is 84.59% reached at the 5<sup>th</sup> epoch.

We can prove that the adoption of the multimodal model to increase images classification scores is successful, considering that accuracy on images only is 71,67% while accuracy on images + text is 84,59%. However, this is not true for

text classification: adding images to related texts leads to an improvement in classification lesser than 0,5%, at the cost of increased complexity, training time and resources needed. We test different merging layers (multiplication, sum,

TABLE VI: Comparison of different merging layers for the late fusion approach represented in Fig. 1

Test Accuracy on UPMC Food-101		
Merging Layer	Test Acc.	Test Loss
Sum	79,40%	1,24
Maximum	79,19%	2,08
Weighted Ensemble	84,58%	1,00
Concatenation	<b>84,59%</b>	1,01

weighted ensemble, maximum), to have a comparison with the concatenation, obtaining the results visible in Table VI. It is possible to observe that the sum and maximum layers are similar in the accuracy results (79%), but do not reach the level of accuracy of the concatenation and weighted ensemble, whose performances are analogous (84,59% and 84,59%, respectively).

As for the late fusion model, we train the early fusion of Table IV using SGD optimizer, reducing the learning rate from 0.01 to 0.001 and finally to 0.0001 when no improvement in test accuracy is observed for 2 consecutive epochs. Looking at Figure 4, differently from what seen with the late fusion multimodal approach, the early fusion reaches a satisfactory result: 92,50% accuracy on test set. As visible in Table VII, the early fusion approach outstands both the single classifiers and the late fusion approach, increasing the classification accuracy of almost eight percentage points.

TABLE VII: Comparison of the models for the classification of UPMC Food-101 [4].

Test Accuracy on UPMC Food-101			
Images Model	Text Model	Late Fusion	Early Fusion
71,67%	84,41%	84,59%	<b>92,50%</b>

We show the predictions of the three models (Inception-based, BERT + LSTM and early-fusion multimodal) on the test dataset, reporting as follow:

- 1) **image** (path to the classified image)
- 2) **actual\_class** (to which the image and the related text belong)
- 3) **bert\_prediction** (prediction made by the BERT + LSTM model on text only)
- 4) **inception\_prediction** (prediction made by the Inception-based model on image only)
- 5) **multimodal\_prediction** (prediction made by the multimodal early-fusion model on both image and text)

Figure 5 shows some examples of correct classifications made by the early fusion model on test images and their related text.

In our analysis, we first consider some cases in which the classification by the Inception-based model on image is

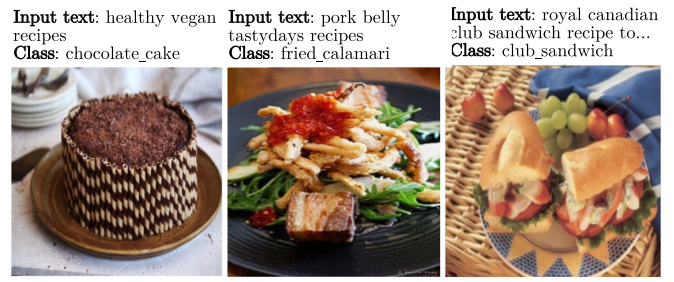


Fig. 5: Examples of correct classifications made by the early fusion model.

wrong, but the prediction made on both image and text by the multimodal classifier is correct (these are then cases in which the text helps making a correct prediction).

Figure 6 shows a few of these cases, randomly selected.

It is possible to notice that the two examples of Figure 6 are noisy images, or either are difficult to classify even for a human person.

	Text	Visual	Fusion	Input image
Actual:	nachos	nachos	nachos	
Predicted:	nachos	omelette	nachos	
Input text:	The best pizza nachos recipes   For Women - Part 2			nachos_411.jpg
Actual:	beignets	beignets	beignets	
Predicted:	beignets	cup_cakes	beignets	
Input text:	Mardi gras beignets without deep frying – National ...			beignets_785.jpg

Fig. 6: Examples in which the classification on images only is wrong but the classification on image and text is correct.

We then consider some cases in which the classification made by the BERT + LSTM model on text is wrong, but the prediction on both image and text by the multimodal classifier is correct (these are cases in which the image helps improving the prediction, contrarily to the previous cases).

	Text	Visual	Fusion	Input image
Actual:	lasagna	lasagna	lasagna	
Predicted:	ravioli	lasagna	lasagna	
Input text:	Campbell's Shortcut Ravioli Lasagna Recipe			lasagna_731.jpg
Actual:	paella	paella	paella	
Predicted:	onion_rings	paella	paella	
Input text:	Onion Rings recipe			paella_178.jpg

Fig. 7: Examples in which the classification on text only is wrong but the classification on image and text is correct

Figure 7 contains some samples of these classifications. It is possible to observe that the texts may contain the names of multiple classes, inducing the classifier in the wrong direction (see 'lasagna\_731.jpg'), or even contain only the names of wrong classes (see 'paella\_178.jpg' that includes the word 'Onion Rings' instead of 'Paella').

Thankfully, adding the image helps the multimodal classifier in correctly classifying this kind of samples.

Finally, we consider the cases in which the classification by BERT + LSTM model on text is wrong, the classification by the Inception-based model on the image only is wrong, but

the classification on both text and image by the multimodal early-fusion model is correct. These are examples in which only the combination of two sources of information allows to make a correct prediction.



	Text	Visual	Fusion	Input image
Actual:	breakfast_burrito	breakfast_burrito	breakfast_burrito	
Predicted:	croque_madame	omelette	breakfast_burrito	
Input text:	Breakfast Recipes on Pinterest			breakfast_burrito_865
Actual:	falafel	falafel	falafel	
Predicted:	french_onion_soup	hummus	falafel	
Input text:	Good Chef Bad Chef - Recipe Detail			falafel_194.jpg

Fig. 8: Examples in which the classifications on text-only is wrong, on the image only is wrong, but on both image and text (multimodal) is correct.

Figure 8 contains random examples of such cases. We can note that the images are not 100% clear, and the texts alone also do not suggest a correct prediction. However, combining images with texts lead to the correct classification.

We compare our results with the ones available in literature for the UPMC Food-101 multimodal dataset.

Wang *et al.* [4] finally achieve 40.2% classification accuracy on images, 82% accuracy on texts and 85.1% accuracy on the fusion of both. Our approach overcomes their results in every modal.

Kiela *et al.*'s [12] results on the UPMC Food-101 are 56.7% accuracy on images, 88% accuracy on text and 90.8% accuracy on the fusion of both. They obtain a better result with text only modal, but our approach overcomes their results in image and fusion.

Narayana *et al.* [16] propose a novel method called HUSE, "Hierarchical Universal Semantic Embeddings". They obtain outstanding results, in particular 73.8% classification accuracy on images, 87.3% on text and 92.3% on the fusion of both. In this case their results into the single modal are better than ours, but our fusion proposal have better accuracy than their proposal. The comparison between all the results is shown in Table VIII.

TABLE VIII: Comparison of classification results available in literature on UPMC Food-101

	Image	Text	Fusion
Wang <i>et al.</i> [4]	40.2	82	85.1
Kiela <i>et al.</i> [12]	56.7	<b>88</b>	90.8
Narayana <i>et al.</i> [16]	<b>73.8</b>	87.3	92.3
Proposed	71.67	84.6	<b>92.5</b>

## VI. CONCLUSION

Multimodal classification is becoming increasingly of interest, especially due to a large amount of multimodal data available on the internet. To be able to manage this type of data we need efficient models. In this paper, we compared two multimodal fusion methods adapted with stacking techniques and found a model that allows us to achieve maximum accuracy and overcome the state of the art on the UPMC Food-101 dataset. Kiela *et al.* [12] and Narayana *et al.* [16] showed

that to have a better multimodal result at least one of the single modal parts must reach the state of the art. In our proposal we show how by combining two models that achieve a good result, we can overcome the state of the art. We hope this work will serve as a useful basis for further work on multimodal classification and for this reason we are also making the code and data we used publicly available.

As future work, we can test this approach with other multimodal datasets with text and image or with different modalities such as audio.

## REFERENCES

- [1] I. Gallo, A. Calefati, and S. Nawaz, "Multimodal classification fusion in real-world scenarios," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 05, pp. 36–41, 2017.
- [2] S. Nawaz, A. Calefati, M. Caraffini, N. Landro, and I. Gallo, "Are these birds similar: Learning branched networks for fine-grained representations," in *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–5, IEEE, 2019.
- [3] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.
- [4] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, IEEE, 2015.
- [5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [11] MatConvNet, "Pretrained models," 2020. <https://www.vlfeat.org/matconvnet/pretrained/>.
- [12] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, "Efficient large-scale multi-modal classification," *arXiv preprint arXiv:1802.02892*, 2018.
- [13] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [15] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [16] P. Narayana, A. Pednekar, A. Krishnamoorthy, K. Sone, and S. Basu, "Huse: Hierarchical universal semantic embeddings," *arXiv preprint arXiv:1911.05978*, 2019.
- [17] Keras, "Keras applications," 2020. <https://keras.io/api/applications/>.
- [18] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101—mining discriminative components with random forests," in *European conference on computer vision*, pp. 446–461, Springer, 2014.