DeepFusion: Deep Ensembles for Domain Independent System Fusion

Mihai Gabriel Constantin[1], Liviu-Daniel Ştefan[1], Bogdan Ionescu[1]

University "Politehnica" of Bucharest, Romania
mihai.constantin84@upb.ro

Abstract. While ensemble systems and late fusion mechanisms have proven their effectiveness by achieving state-of-the-art results in various computer vision tasks, current approaches are not exploiting the power of deep neural networks as their primary ensembling algorithm, but only as inducers, i.e., systems that are used as inputs for the primary ensembling algorithm. In this paper, we propose several deep neural network architectures as ensembling algorithms with various network configurations that use dense and attention layers, an input pre-processing algorithm, and a new type of deep neural network layer denoted the Cross-Space-Fusion layer, that further improves the overall results. Experimental validation is carried out on several data sets from various domains (emotional content classification, medical data captioning) and under various evaluation conditions (two-class regression, binary classification, and multi-label classification), proving the efficiency of DeepFusion.

Keywords: ensemble learning · deep neural networks · deep ensembles

1 Introduction

Ensemble systems have demonstrated to be effective models for solving a large variety of problems in several domains, including image [14] and video [28] classification, speech recognition [6], and broader data stream processing [11] to name but a few. *Ensemble learning* is a universal term for methods that employ a pool of *inducers* to generate predictions, typically in supervised machine learning tasks. Despite the current advances in knowledge discovery, single learners did not obtain satisfactory performances when dealing with complex data, such as class imbalance, high-dimensionality, concept drift, noisy data, etc. In this context, ensemble learning tries to fill this gap by obtaining a strong learner based on the experience of the myriad of classifiers it incorporates. Selecting an appropriate set of inducers according to the classification problem for obtaining an accurate ensemble is still an open research problem [24, 11]. As ensemble accuracy is governed by the law of large numbers, one must consider the relationships between *accuracy* and *diversity* of the inducers within an ensemble. Here, diversity refers to the capability of the inducers of responding differently to the same input. This paradigm is supported by the no-free-lunch theorem

formulated by Wolpert [26]. Concretely, given a learning task, where N inducers are independently trained on a specific data set, it is improbable that the errors from these N individual learners are completely uncorrelated. However, it is more probable that the ensemble accuracy can surpass the average accuracy of its members if they promote high diversity in predictions. Finding the optimal accuracy-diversity trade-off has been actively studied [19], showing that ensemble error decreases as base model error decreases and diversity increases. Nevertheless, the pioneering work in [17] shows that the relationships between accuracy and diversity of the members in an ensemble are more complex, and it is not guaranteed that enhancing ensemble diversity improves the predictive results. This raises an essential technical research question: *How to create high accuracy ensembles to account for all the aforementioned factors?* To answer this question, we aim to model the bias learned by each classifier in the ensemble setting and the correlations between the biases via a consensus learning method, to perform retrieval robustly, and improve the performance of the inducers.

Several methods have been used in the literature, starting from AdaBoost [8] with it's variants, e.g., soft margin AdaBoost [22] and Gradient boosting machines [10] with improved iterations as XGBoost [4], to Bagging [2] and Random Forests [3]. Despite the success of DNNs in recent years, there is currently little literature on the integration of ensemble learning in deep neural networks. Recent works use ensembles to boost over features [7], where deep neural networks are trained, in an entangled setting, stacking diverse data representations. To the extent of our knowledge, DNNs have not been explored as *ensembling learners*, where the input is represented by the prediction of a plethora of classifiers. One of the reasons is that the popular CNN layers are designed to learn patterns based on how the data is organized in space, whereas in ensembles the pixels are substituted with predictions of different classifiers that are not correlated.

In this context, the contribution of this work is 5-fold: (i) we introduce a set of novel ensembling techniques that use deep neural networks; (ii) we further enhance the performance of our deep ensembling methods by adding attention layers to the networks, with the particular role of filtering the input created by the inducer systems; (iii) we introduce a novel input pre-processing method, that groups the data according to the overall correlation between inducers, allowing for a better analysis of the input and for the introduction of more complex processing layers; (iv) we introduce a novel deep learning layer, called Cross-Space-Fusion layer that optimally utilizes the new pre-processed data and further enhances the results of the deep ensembling methods; (v) the proposed approaches are not dependent on a particular category of learning tasks or data. Evaluation is carried out on a set of diverse tasks, targeting several types of machine learning problems: two-class regression, binary classification, and multi-label classification. We achieve state-of-the-art performance compared to other traditional ensembling techniques and best performers from the literature. To allow reproducibility, we publish the code on GitHub[1]. The code is developed in Python 3 and is tested on the Keras 2.2.4 and Tensorflow 1.12 libraries.

[1] https://github.com/cmihaigabriel/DeepFusionSystem_v2

2 Proposed method

For a standard ensembling system, given a set S of M samples and a set F of N classifier or regression inducer algorithms, each algorithm will produce an output for every given sample $y_{i,j}, i \in [1, N], j \in [1, M]$, as follows:

$$\begin{cases} S = \begin{bmatrix} s_1 & s_2 & ... & s_M \end{bmatrix} \\ F = \begin{bmatrix} f_1 & f_2 & ... & f_N \end{bmatrix} \end{cases} \Rightarrow Y = \begin{bmatrix} y_{1,1} & \cdots & y_{1,M} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ y_{N,1} & \cdots & y_{N,M} \end{bmatrix} \tag{1}$$

Ensembling systems involve the creation of a new aggregation algorithm E, that can detect and learn patterns in a training set composed of image or video samples, inducers, and their outputs, and apply those patterns on a validation set to produce a new output for every sample, e_i, that is a better prediction of the ground truth values of the validation set. Furthermore, a generalized ensembling system must take into account the type of output required by the studied problem. Therefore, while in binary classification tasks, the output values are $e_i \in \{0, 1\}$ and in regression tasks the values are $e_i \in [0, 1]$ or $[-1, 1]$, for multi-label prediction, e_i is actually represented by a vector of values for each of the L labels assigned to the dataset. With this in mind, we propose the DeepFusion approach – to deploy several DNN architectures that take as input a set of inducer predictions and produce new outputs e for the given samples. Our assumption is that deep architectures will be able to learn the biases of different inducers, no matter how high or low their performance is. We therefore propose three different types of network architectures: dense networks (Section 2.1), attention augmented dense architectures (Section 2.2) and finally, dense architectures augmented with a novel type of layer, called Cross-Space-Fusion layer (Section 2.3).

2.1 Dense networks

Given their ability to classify input data into output labels correctly, dense layers have represented an integral part of deep neural networks in many domains. In our particular case, we use a set of dense layers for combining the inputs of all inducers and, in the training phase, learn the biases of the inducer systems and adapt the internal parameters of the dense layers in a manner that will allow the prediction of validation data. Considering that dense layers need no special assumption with regards to the nature of the input, we believe using them is also useful for creating a domain-independent ensembling system. Figure 1 presents the diagram of the dense network approach.

To optimize the results of our dense networks, we start by building a system that searches for the best parameters of these networks. We accomplish this by varying several parameters of the dense network: (i) the optimum width by testing the following number of neurons per layer: {25, 50, 500, 1000, 2000, 5000}; (ii) the depth of the network by changing the number of dense layers, testing the following values: {5, 10, 15, 20, 25}; (iii) adding or removing batch normalization layers between the dense layers. We validate these architectures

in our experiments and change these parameters until the best architecture with regards to prediction capabilities is detected.

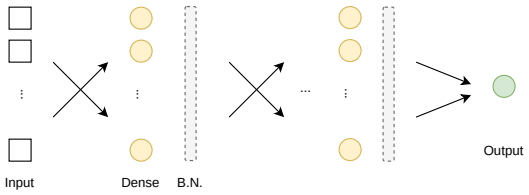

Fig. 1: DeepFusion dense network architecture (DF-Dense): variable number of layers, number of neurons per layer and the presence or absence of BN layers.

2.2 Dense networks with Attention layers

While in the previous step we try to discover the best dense network, in this section we focus on boosting the predictive power of our approach by inserting soft-attention estimators after the last dense layer in the structure. Our intuition is that such mechanisms will learn to attend to specific informative features, by unveiling the relationships between individual classifiers. We denote the input vector as x_i, the soft attention vector as a_i with $a \in [0, 1]$, the learned attention mask $\hat{a}_i$ is then computed by element-wise multiplication of input vector and the attention mask: $\hat{a}_i = a_i \odot x_i$. Finally, the attention mask is learned in a supervised fashion with back-propagation. The pipeline of our proposed approach called DF-Attn is depicted in Figure 2.

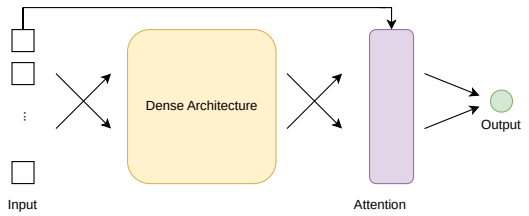

Fig. 2: DF-Attn architecture: attention masks automatically discover mutually complementary individual classifiers and disable the counterproductive ones.

2.3 Dense networks with Cross-Space-Fusion layers

The final architecture augmentation is represented by a novel type of layer, the Cross-Space-Fusion (*CSF*) layer, that can exploit spatial correlations, and several input pre-processing (*decoration*) techniques that allow the implementation

of the CSF layer. While the introduction of convolutional layers has greatly improved the performance of deep neural networks [16] by allowing the spatial processing of data, several aspects hinder the use of such layers. First of all, given the input matrix Y, with the shape presented in Equation 1, there is no intrinsic spatial information associated with such an input; therefore, some input decoration techniques should be used to generate these spatial correlations.

To generate spatial information, we choose to pre-process the input data and decorate each element with output scores and data from the most similar inducers. Given an image or video sample $s_i, i \in [1, M]$, each of the N inducer algorithms will produce a set of scores, $Y_i = \begin{bmatrix} y_{1,i} & y_{2,i} & ... & y_{N,i} \end{bmatrix}$, and, as mentioned before, this kind of input has no intrinsic spatial correlation associated with it. In the first step of the input decoration technique, we analyze the correlation between the individual inducers $f_i, i \in [1, N]$. This correlation can be determined by any standard method, such as Pearson's correlation score, but, to ensure an optimized learning process we will use the same metric as the task we are processing. Given any $f_i, i \in [1, N]$ inducer system, that produces the vector $\bar{f_i} = \begin{bmatrix} \bar{f}_1 & \bar{f}_2 & ... & \bar{f_M} \end{bmatrix}$ of outputs across the entire set of samples, and a vector of correlation scores $cr_i = \begin{bmatrix} cr_{1,i} & cr_{2,i} & ... & cr_{N-1,i} \end{bmatrix}$ between this inducer and all the other inducers can be generated. We then create the following structures:

$$C_{i,j} = \begin{bmatrix} c_{1,i,j} & c_{2,i,j} & c_{3,i,j} \\ c_{8,i,j} & s_{i,j} & c_{4,i,j} \\ c_{7,i,j} & c_{6,i,j} & c_{5,i,j} \end{bmatrix}, R_{i,j} = \begin{bmatrix} r_{1,i,j} & r_{2,i,j} & r_{3,i,j} \\ r_{8,i,j} & 1 & r_{4,i,j} \\ r_{7,i,j} & r_{6,i,j} & r_{5,i,j} \end{bmatrix} \tag{2}$$

In this example, each element $s_{i,j}$, representing the prediction produced by inducer i for sample j, is decorated with scores from the similar systems, $c_{1,i,j}$ representing the most similar system, $c_{2,i,j}$ representing the second most similar system and so on. For the second dimension of our new matrix we input the correlation scores for the most similar system ($r_{1,i,j}$), the second most similar ($r_{2,i,j}$) and so on, with the value 1 as centroid, corresponding to the initial $s_{i,j}$ element. Finally, the new input of our deep ensembling models is represented by a 3-dimensional matrix, composed of $M \times N$ $C_{i,j}$-type 3×3 bi-dimensional elements and the same number of $R_{i,j}$-type 3×3 bi-dimensional elements. We denote the entire input matrix, composed of these structures with I_{proc} and its size is $(3 \times N, 3 \times M, 2)$. After pre-processing the input I_{proc}, it is fed into the CSF layer. For each group of centroids (C_i, R_i), the neural network learns a set of parameters $\alpha_{k,i}$ and $\beta_{k,i}$ that will process each sample as following, therefore combining the elements in each centroid:

$$\begin{bmatrix} \frac{\alpha_{1,i} * s_i + \beta_{1,i} * c_{1,i} * r_{1,i}}{2} & \frac{\alpha_{2,i} * s_i + \beta_{2,i} * c_{2,i} * r_{2,i}}{2} & \frac{\alpha_{3,i} * s_i + \beta_{3,i} * c_{3,i} * r_{3,i}}{2} \\ \frac{\alpha_{8,i} * s_i + \beta_{8,i} * c_{8,i} * r_{8,i}}{2} & s_i & \frac{\alpha_{4,i} * s_i + \beta_{4,i} * c_{4,i} * r_{4,i}}{2} \\ \frac{\alpha_{7,i} * s_i + \beta_{7,i} * c_{7,i} * r_{7,i}}{2} & \frac{\alpha_{6,i} * s_i + \beta_{6,i} * c_{6,i} * r_{6,i}}{2} & \frac{\alpha_{5,i} * s_i + \beta_{5,i} * c_{7,i} * r_{5,i}}{2} \end{bmatrix} \tag{3}$$

The number of parameters used by the CSF layer per centroid pair is 16, thus generating $16 \times M$ parameters that are trained, where M is the total number of inducers. The output of the CSF layer is finally processed by Average Pooling

layers, thus generating a single value for each (C_i, R_i) centroid group and, thus, outputting the same sized matrix as the input before the pre-processing step. We test two setups with regard to data processing. In the first setup, denoted *8S*, all the 8-most similar inducer values are populated in I_{proc}, while in the second setup, denoted *4S*, only the 4-most similar ones are populated. A diagram of the Cross-Space-Fusion architecture is presented in Figure 3.

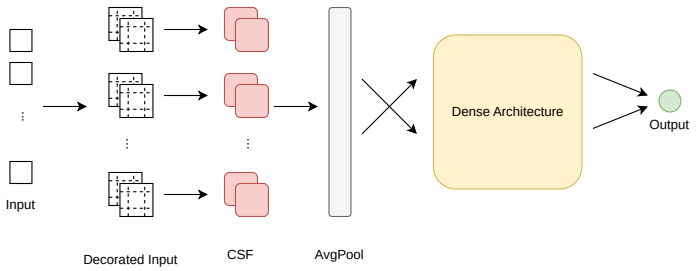

Fig. 3: Cross-Space-Fusion augmented architecture: initial pre-processing steps and the architecture of the entire network (DF-CSF).

3 Experimental setup

3.1 Training protocol

To perform the ensembling of the systems we search for the best performing dense architecture by using the setup presented in Section 2.1. Results are tested against the validation set, and the best performing dense architecture is then augmented with attention layers in the third step and with Cross-Space-Fusion layers in the fourth step. The input is modified for the use of the CSF layers, as described in Section 2.3. As presented in Section 2.3, the CSF layers are applied directly to the dense architecture, and not to its attention augmented variant. For each network, training is performed for 50 epochs, with a batch size of 64, and mean squared error loss for the regression experiments and binary crossentropy for the classification and labeling experiments. We use an Adam optimizer [12], with an initial learning rate of 0.01.

3.2 Data sets

We empirically verify the performance of our approaches, by conducting the experiments on two data sets, namely: the Emotional Impact of Movies [5], and the ImageCLEFmed Concept Detection [20]. These data were validated during the yearly MediaEval[2] and ImageCLEF[3] benchmark initiatives, for multimedia evaluation, and cross-language annotation and retrieval of images, respectively.

[2] http://www.multimediaeval.org/
[3] https://www.imageclef.org/

The MediaEval 2018 Emotional Impact of Movies [5] is a data set for automatic recognition of emotion in videos, in terms of valence, arousal, and fear. Out this regard, the data set offers annotations for two tasks, namely (i) valence and arousal prediction (denoted *Arousal-Valence*), a two-class regression task, consisting of 54 training/validation movies with a total duration of approx. 27 hours, and 12 testing movies with a total duration of approx. 9 hours, and (ii) fear detection (denoted *Fear*), a binary classification task, consisting of 44 training/validation movies, with a total duration of more than 15 hours, and 12 testing movies with a total duration of approx. 9 hours. We considered all the systems participating in the benchmarking campaign, namely 30 systems for the valence and arousal prediction task, and 18 systems for the fear detection task. Inducers ranged from linear SVR/SVM, clustering, multi-layer perceptron, to bidirectional LSTMs, temporal convolutional networks, and ensembles. For a detailed description, the reader may access the participants working notes here[4].

The ImageCLEFmed 2019 Concept Detection (denoted *Caption*) is a multi-label classification image captioning and scene understanding data set [20] consisting of 56,629 training, 14,157 validation, and 10,000 test radiology examples with multiple classes (medical concepts) associated with each image, extracted from the PMC Open Access [23] and Radiology Objects in COntext [21]. In total, there are 5,528 unique concepts, whereas the distribution limits per images in the training, validation, and test sets is between 1-72, 1-77, and 1-34 concepts, respectively. We have considered all the systems participating in the benchmarking campaign, namely 58 systems with inducers ranging from clustering, logistic regression, to ResNet-101, LSTM, CheXNet and ensembles. For a detailed description, the reader may access the participants working notes here[5].

3.3 Evaluation

Ensemble accuracy is governed by the law of large numbers, typically requiring tens of systems to provide learning guarantees and significantly boost the performance. However, they are not used in practice, as it is impossible to implement or retrieve so many systems from the authors, or even to re-run them in identical conditions. In addition, there is no general criterion or best practices in this respect in the literature. In this context, current approaches use a limited number of inducers, e.g., less than 10 [18], which most likely are not representative for a full-scale experiment. To overcome this issue, we have incorporated all the systems developed and submitted in the respective benchmarking competitions and performed all the experiments solely on the test data, as provided by the task organizers, in a repeated k-fold cross validation manner. In this regard, the split samples are randomized, and 100 partitions are generated to assure a thorough coverage of the data, using two protocols: (i) 75% training and 25% testing (KF75), and (ii) 50% training and 50% testing (KF50).

[4] http://ceur-ws.org/Vol-2283/
[5] http://ceur-ws.org/Vol-2380/

Performance evaluation is carried out via the official metrics released by the authors of the data, namely: (i) for the MediaEval 2018 Emotional Impact of Movies data set, we use the Mean Square Error (MSE) and the Pearson's Correlation Coefficient (PCC) for the valence and arousal prediction task, with MSE being the primary metric, and Intersection over Union (IoU) of time intervals, for the fear detection task, and (ii) for the ImageCLEFmed 2019 Concept Detection, the F1-scores computed per image and averaged across all test images. The metrics are computed as average values over all the partitions.

4 Results and discussion

This section presents and analyzes the results of the proposed DeepFusion methods (Section 2) with regards to the three tasks described in Section 3.2. We use the best-performing systems submitted at their respective benchmarking competitions as baselines for comparing our methods, and these methods also represent the best performing inducer systems for our DeepFusion approach.

For the Arousal-Valence data, the best performing method is developed by Sun et al. [25], using a traditional late fusion approach, with results of MSE = 0.1334 and PCC = 0.3358 for arousal and MSE = 0.0837 and PCC = 0.3047 for valence. For the Fear data, the best performing system is developed by Yi et al. [27], using a series of convolutional networks, and achieving a IoU result of 0.1575. Finally, for the Caption data, the best results are achieved by Kougia et al. [15], with a deep learning system that achieves an F1 score of 0.2823.

Another group of baseline methods is represented by classical ensembling approaches. We test several traditional late fusion strategies such as: weighted fusion (denoted LFweight), based on the ranking of individual inducers, taking the maximum inducer score value (LFmax) and taking the average and median values of inducer scores (LFavg, LFmed) for the regression tasks, and max voting (LFvote) for the classification tasks [13]. Furthermore, we also add two boosting mechanisms: AdaBoost [9] (BoostAda) and Gradient Boosting [10] (BoostGrad). For these traditional late fusion approaches we choose inducer combinations that maximize their performance.

4.1 Ablation studies

Network size heavily influences the performance of the network. In cases when the network is too large, due to a lack of sufficient training samples or due to low dimensionality of the inputs, the network can encounter problems such as the exploding gradient problem [1]. Figure 4 presents such a case, studied on the Arousal-Valence data. The arousal and valence graphs represent a network setup of 5 dense layers and a variation of network width. It is interesting to observe that, for numbers of neurons that surpass 1,000, the network is not able to learn the inducer biases, and the MSE results are lower than the state-of-the-art (SOA) results. Furthermore, in those particular cases, the value of the PCC metric approaches 0, denoting that perhaps the network outputs values

close to neutral for the validation samples [5]. Also, the addition of BN layers contributed to the performance of the arousal detection network, allowing for a transition to a larger network, from 50 neurons/layer, to 500.

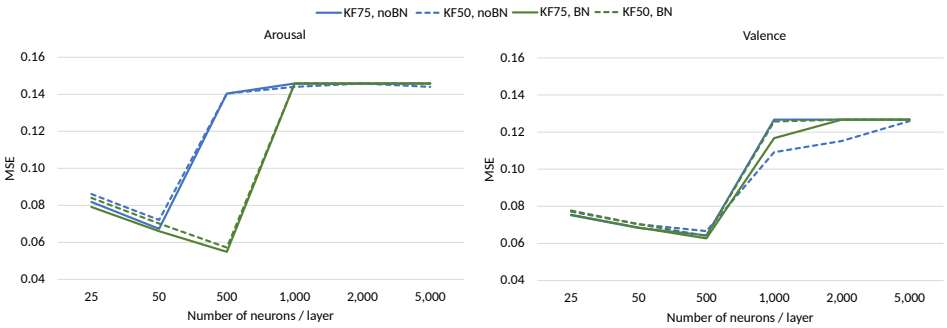

Fig. 4: Ablation study with regards to the influence of network depth and batch normalization layers, on the Arousal-Valence data.

Finally, it is worth noting that, for the DF-CSF architecture, there are differences with regards to the optimal input decoration mechanism. While for the regression data, Arousal-Valence, the best performing decoration scheme uses only 4 similar systems for decorating each input sample (4S), for the classification and multi-labeling data, Fear and Caption, the best performances for the DF-CSF architectures uses all the 8 most similar systems in decorating the input samples (8S). The difference in results is most evident for the Fear data, where the DF-CSF-8S architecture has an IoU result of 0.2242, while the DF-CSF-4S architecture has a result of 0.2173. The difference between the optimal approach with regards to the decoration mechanism proves that for different approaches and tasks, different decoration mechanisms must be employed and that newer variations of our mechanism may provide even better results.

4.2 Results

The results of our proposed DeepFusion architectures are presented in Table 1. As expected, training on a KF75 split produces better results than KF50, but in both scenarios the results clearly surpass the state-of-the-art and the traditional late fusion approaches. It is important to note that the results for the traditional late fusion approaches did not improve state-of-the-art results by a large margin, and in some cases, performed worse than the SOA systems. Overall the best performing architecture is represented by the CSF layer augmented networks.

For the Arousal-Valence data, the best performing DF-Dense architecture is composed, for both arousal and valence, of a 5 layer network, with each layer having 500 neurons and uses batch normalization layers between the dense layers. Even though the state-of-the-art results were high with regards to MSE [25],

Table 1: Results for the three datasets: Arousal-Valence, with MSE and PCC metrics, Fear, with IoU metric and Caption, with F1-score metric. For reference, we present the best performing state-of-the-art systems [25, 27, 15] (SOA) and the traditional late fusion mechanisms. We present the best performing dense (DF-Dense), attention (DF-Attn) and Cross-Space-Fusion (DF-CSF) architectures. The best results for each type of data split are presented (KF75 and KF50), including the original (orig.) split for the SOA systems and the best performing overall architectures for each split are presented in bold.

System	Split	Arousal (MSE)	Arousal (PCC)	Valence (MSE)	Valence (PCC)	Fear (IoU)	Caption (F1)
SOA	orig	0.1334	0.3358	0.0837	0.3047	0.1575	0.2823
LFweight	orig	0.1297	0.2819	0.0831	0.3011	-	-
LFmax	orig	0.1374	0.3135	0.0851	0.2563	-	-
LFavg	orig	0.1316	0.3347	0.0821	0.2916	-	-
LFmed	orig	0.1310	0.3340	0.0820	0.2902	-	-
LFvote	orig	-	-	-	-	0.1381	0.2511
BoostAda	KF75	0.1253	0.3828	0.0783	0.4174	0.1733	0.2846
BoostGrad	KF75	0.1282	0.3911	0.0769	0.3972	0.1621	0.2834
DF-Dense	KF75	0.0549	0.8315	0.0626	0.8101	0.2129	0.3740
	KF50	0.0571	0.8018	0.0640	0.7876	0.1938	0.3462
DF-Attn	KF75	0.0548	0.8339	0.0626	0.8107	0.2140	0.3659
	KF50	**0.0568**	0.8036	0.0640	0.7888	0.1913	0.3522
DF-CSF	KF75	**0.0543**	**0.8422**	**0.0625**	**0.8123**	**0.2242**	**0.3912**
	KF50	**0.0568**	**0.8073**	**0.0637**	**0.7903**	**0.2091**	**0.3610**

0.1334 for arousal and 0.0837 for valence, our DF-Dense systems managed to improve those results, reaching to 0.0549 and 0.0626. Both the DF-Attn and DF-CSF networks further improved these results, with the CSF network being the best overall performer; however, the difference in results to DF-Dense was not very substantial. Another interesting analysis can be performed on the PCC results. Here, our proposed networks significantly improve the results, indicating a better understanding of borderline cases, as theorized in the competition overview paper [5]. The state-of-the-art results are improved to a maximum of 0.8422 for arousal and 0.8123 for valence by the DF-CSF network. Regarding the DF-CSF network, the architecture performed best with the 4S setup.

For the Fear data, the optimal DF-Dense configuration is composed of a 10 layer network, with each layer having 500 neurons, and no batch normalization layers. This configuration achieves a final score of 0.2129, increasing the state-of-the-art result by 35.17%. Furthermore, both the DF-Attn and the DF-CSF configurations increase this score, with a maximum IoU of 0.2242, representing a 42.35% increase over the original performance, for an 8S setup of DF-CSF.

Finally, for the Caption data, the best performing DF-Dense configuration is represented by a 5 layer network, with 500 neurons per layer and no batch normalization. This configuration achieves an F1-score of 0.3740, increasing the state-of-the-art results by 32.48%. Interestingly, for the KF75 split, the DF-Attn

network does not perform better than the DF-Dense network. However, similar to the other datasets, the best performing system is the DF-CSF, with an F1-score of 0.3912, under the 8S setup.

5 Conclusions and future work

In this paper, we present a set of novel ensembling techniques, that use DNN architectures as the primary ensembling learner. Our architectures integrate dense and attention layers, but also a novel input decoration technique, that creates spatial information out of inducer outputs and a novel deep neural network layer, the Cross-Space-Fusion layer, that is able to manipulate the newly created spatial information. We evaluate our approaches on three tasks from diverse domains (emotional content processing and medical image captioning) and with diverse problem formulations (two-class regression, binary classification, and multi-label classification). We validate our results by comparing them with the current state of the art on the three tasks and with traditional ensembling approaches. Results show significant improvements, by a margin of at least 38.58%, therefore validating our DeepFusion methods and techniques.

Given the encouraging results presented in this paper, future developments of our methods may prove useful for solving even more tasks under different problem formulations. Considering the usefulness of our input decoration method, some further work towards improving the results may include incorporating a larger set of decoration methods, or adding both similar and dissimilar inducers in the decoration scheme. We also believe that further work on the Cross-Space-Fusion layer may provide even better results.

Acknowledgements This work was funded under project SMARTRetail, agreement 8PTE/2020, grant PN-III-P2-2.1-PTE-2019-0055, Ministry of Innovation and Research, UEFISCDI and AI4Media "A European Excellence Centre for Media, Society and Democracy", grant #951911, H2020 ICT-48-2020.

References

1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks **5**(2), 157–166 (1994)
2. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
3. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
5. Dellandréa, E., Huigsloot, M., Chen, L., Baveye, Y., Xiao, Z., Sjöberg, M.: The mediaeval 2018 emotional impact of movies task. In: MediaEval (2018)
6. Deng, L., Platt, J.C.: Ensemble deep learning for speech recognition. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)

7. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)
8. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence **14**(771-780), 1612 (1999)
9. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. pp. 23–37. Springer (1995)
10. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
11. Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A.: A survey on ensemble learning for data stream classification. ACM Computing Surveys **50**(2), 1–36 (2017)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. IEEE transactions on pattern analysis and machine intelligence **20**(3), 226–239 (1998)
14. Kontschieder, P., Fiterau, M., Criminisi, A., Rota Bulo, S.: Deep neural decision forests. In: Proceedings of the IEEE international conference on computer vision. pp. 1467–1475 (2015)
15. Kougia, V., Pavlopoulos, J., Androutsopoulos, I.: Aueb nlp group at imageclefmed caption 2019. In: CLEF2019. CEUR Workshop Proceedings. pp. 09–12 (2019)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
17. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine learning **51**(2) (2003)
18. Li, X., Huo, Y., Jin, Q., Xu, J.: Detecting violence in video using subclasses. In: ACM Conference on Multimedia Conference, 2016. pp. 586–590 (2016)
19. Liu, L., Wei, W., Chow, K.H., Loper, M., Gursoy, E., Truex, S., Wu, Y.: Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. arXiv preprint arXiv:1908.11091 (2019)
20. Pelka, O., Friedrich, C.M., García Seco de Herrera, A., Müller, H.: Overview of the imageclefmed 2019 concept detection task. CLEF working notes, CEUR (2019)
21. Pelka, O., Koitka, S., Rückert, J., Nensa, F., Friedrich, C.M.: Radiology objects in context (roco): A multimodal image dataset. In: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, pp. 180–189. Springer (2018)
22. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for adaboost. Machine learning **42**(3), 287–320 (2001)
23. Roberts, R.J.: Pubmed central: The genbank of the published literature (2001)
24. Sagi, O., Rokach, L.: Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**(4), e1249 (2018)
25. Sun, J.J., Liu, T., Prasad, G.: Gla in mediaeval 2018 emotional impact of movies task. In: Proc. of MediaEval 2018 Workshop (2018)
26. Wolpert, D.H.: The supervised learning no-free-lunch theorems. In: Soft computing and industry, pp. 25–42. Springer (2002)
27. Yi, Y., Wang, H., Li, Q.: Cnn features for emotional impact of movies task. In: Proc. of MediaEval 2018 Workshop (2018)
28. Zhou, Z.H., Feng, J.: Deep forest: Towards an alternative to deep neural networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 3553–3559 (2017)