Fusing Visual Features and Metadata to Detect Flooding in Flickr Images

Rabiul Islam Jony[1], Alan Woodley[2], Dimitri Perrin[3]
School of Computer Science, Science and Engineering Faculty[1,2,3]
Queensland University of Technology
ARC Centre of Excellence for Mathematical & Statistical Frontiers[2]
Brisbane, Australia
{r.jony, a.woodley, dimitri.perrin}@qut.edu.au

***Abstract*—Social media platforms such as Flickr have become a source of information for the assessment of natural disasters, for instance assisting in flood mapping. Visual features and textual metadata have been used to identify natural disasters in social media images, however, they have often been used separately. Here, we fuse these two modes together using two fusion methods and deep learning to identify flood images in the MediaEval 2017 dataset. A novel backpropagation technique, Direct Backpropagation (DBP) is used to train a neural network for the classification. The results show that the fusion methods improve the classification accuracy compared to their individual counterparts. We compare our proposed learning method with other baseline methods and find it producing highest classification results. For external evaluation, the results are compared with MediaEval 2017 methods, where our methods outperform most of them.**

***Index Terms*—Image classification, Multimodal fusion, Social media, Deep learning, Information retrieval**

I. Introduction

Our world is in a continuous threat of natural disasters that can cause thousands of deaths, billion dollars of damage, and destroy natural landmarks [1]. Research shows that proper assessment of these disasters can reduce the damage they cause. For fifty years, remote sensors such as satellites and radars have been collecting images of natural disasters that have been used by researchers for assessment [2]–[4]. However, due to long revisit time and cloud cover, remote sensing images are often unavailable [5].

Alternatively, social media such as Twitter, Facebook, Flickr and Instagram are enriched with in-situ information of natural disasters [6]. This information typically appears as geo-tagged photos and associated metadata such as keywords assigned by the user. For example, each Flickr post is a photo entry with user-defined Title, a small Description, user tags, time stamp and location. Research shows that this multimedia information can be employed for different applications including natural disaster detection [7] and assessment [8]. However, social media data are also associated with some limitations such as lack of reliability and uncertainty [10]. Researchers have tested different techniques to identify flooding evidence in social media images. Visual features have been used for such applications. Alternatively, the metadata have been exploited to a limited extent. This is due to the challenges associated with analysing unstructured and noisy text data [8].

Data fusion is a popular technique in remote sensing community [9]. Fusing together of multimodal data typically produce better classification results [13], [14]. Here, we apply two fusion approaches, feature-level fusion and decision fusion, to combine visual features and metadata from social media. These approaches are used to classify Flickr images into two categories, flood and no-flood. We also test the visual features and metadata separately to evaluate the fusion approaches.

We employ a neural network for the classification using the multimodal data described in previous paragraph. However, instead of using typical backpropagation technique for the learning, we propose a novel method named Direct Backpropagaton (DBP).

For evaluation, we use the Disaster Image Retrieval from Social Media (DIRSM) dataset from MediaEval 2017 [15]. We compare our different approaches using classification accuracy. The results show that the visual modes outperform the textual modes and achieve higher accuracy when used separately. However, the fusion approaches further improve the accuracy. The decision fusion approaches perform better than feature-level fusion, but require more computation. We evaluate our proposed learning method by comparing our results with other baseline methods. The results show that our proposed method produce highest training accuracy in minimum time. We also compare our results with other approaches presented in MediaEval 2017 as external evaluation suing Mean Average Precision (MAP), where our fusion approaches are comparable to the most successful approaches.

II. Related Work

A. Classification using visual features

In the last decade, Convolutional Neural Networks (CNN) have achieved high success for visual feature extraction in the image processing and remote sensing domain [16], [20]. CNNs typically require large training datasets and with the development of large standard datasets such as ImageNet [17] and Places [18], they can now be trained more accurately. These datasets contain millions of images labelled in thousands of categories and can be exploited to pre-train CNNs for visual feature extraction. For example, ImageNet was used to pre-train different architectures of CNNs for visual feature extraction in [19]. However, researchers have argued that the

ImageNet dataset mostly contains object type images such as, 'table' and 'tree' and therefore, to extract scene level features such as 'river', and 'garden', it is important to use a dataset that contains scenic type images such as the Places dataset [7]. The Places dataset was tested for pre-training CNNs in [21]. However, the ImageNet dataset contains around twice more images categorized into thousands more classes than the Places dataset [6]. Recently the Places dataset has been updated to Places365-Standard by increasing the number of images per category [18].

Among different CNN architectures, Xception, ResNeXt and InceptionV3 are the top performing models on ImageNet. However, the ResNext model calculates twice the number of parameters compared to other models and therefore, requires more resources. On the Places dataset, VGG16, ResNet and GoogleNet are the top performing models [6].

B. Classification using metadata

There are four categories of typical social media image metadata: textual, geographical, temporal and social media information [22].

Here, we use only the textual metadata, because of challenges such as unavailability [1] and ethics [23] associated with others. Among the textual metadata, user tags that are the keywords assigned by the user have been popular for event detection in social media due to their availability and structured nature [8]. In Flickr the title, which is typically a one sentenced statement, and the Description that is a longer explanation of an image, are unstructured, and sometimes noisy [6].

For textual feature extraction, word embedding and Bag-of-Words (BoW) are two popular techniques [24]. However, for domain specific applications, word embedding can be sensitive to the domain knowledge. Therefore, this technique is suitable only if pre-trained on similar dataset and typically used in such applications [24].

For text classification, several studies relied on a support vector machine (SVM) classifier because of its simplicity and ease of interpretability [8]. However, neural networks are also popular for text classification [25].

C. Multimodal fusion

Two basic multimodal fusion methods are early fusion and late fusion. Early fusion, also known as feature-level fusion, joins the input features from multiple modalities and fuses them together as one feature set [26]. Researchers have tested different approaches such as concatenation [27], linear sum [28], pooling [29], and gated method [30] for feature-level fusion of visual modes and textual modes. Research shows that the concatenation method is most efficient because it can produce competitive results with minimum computation, where other methods such as linear sum require the feature dimensions to be equal [14]. Alternatively, the late fusion method takes the unimodal decision or the prediction values from different modes as the input and calculates the final decision value using a fusion mechanism such as averaging [31], voting [32], weighting [33] and learned model [25]. This method is also known as the decision fusion method.

Both of these methods have their advantages and disadvantages. For example, in decision fusion, the heterogeneity of the features is dealt with suitable individual classifiers for each modality. Therefore, the strength of individual modalities are exploited separately [26]. This is why they typically perform better than feature-level fusion [14], [26]. However, there is no scope to learn and exploit the correlation and the interaction between features of different modalities in decision fusion. It also requires increased number of classifiers and computation.

Alternatively, in feature-level fusion only one classifier is required that takes multimodal features from multiple modalities together as the input pattern [26]. Feature-level fusion is also considered as the true multimedia feature representation because they combine the features at raw level and can exploit the correlation among them [33]. Research shows that depending on the feature and the feature processing method, feature-level fusion can outperform decision fusion method [35]. However, with different types of features having dissimilar capability, the learning becomes challenging as they require sophisticated weight learning and assigning method.

Other interesting multimodal fusion models include multi-view [36], multimodal representation [13], hybrid of decision and feature-level fusion [37], Boltzmann Machines [38] and autoencoders [39]. However, such methods suffer from increased computational complexity [40]. Alternatively, deep learning gained much popularity in this field due to their computationally tractable representation capability and superior performance [41].

D. Deep learning

Artificial neural networks (ANN) have the capability to learn and assign weights to individual input based on their impact on the classification process [42]. Using a feedforward method, they calculate the output and the error, which is also known as the cost function, as the difference between predicted output and target. After that, the error information is fed back in a backpropagation method to recalculate the output with changed parameters. This process of updating the parameters or the weights to reduce the error is known as training. The training process continues until the error is reduced to a certain acceptable level [42].

A typical neural network comprises of three layers: input, hidden and output. Each layer is comprised of multiple neurons and each neuron is connected to next consecutive layer's neurons. These connections have weights that represent the reliability of the signals [42]. Backpropagation (BP) is a popular method to calculate a gradient that updates these weights based on the error signal [42]. However, backpropagation method requires symmetric weights, a separate phase for inference and learning, and the learning signals have to be propagated backward layer-by-layer from the output layer to the input layer. This also means that each layer is dependent on previous layer during the update process, which necessitates a waiting time [43].

Researchers have proposed a few alternative methods of backpropagation. For example, Lillicrap et al., have proposed a Feedback Alignment (FA) method that bypasses random feedback weights and can achieve similar accuracy as BP [44]. Nøkland proposed a modified version of FA, the Direct Feedback Alignment (DFA) [43]. He showed that the feedback to shallow layers need not be propagated at all; only the cost function from last layer can be utilized with the random feedback to calculate the gradient of each layer. He further modified the technique to develop the Indirect Feedback Alignment (IFA) method. In IFA, the first layer weights are updated using the direct feedback. The network is then placed into a forward path again, using the derivative of first layer to update the second layer and so on. These techniques show that the last layer error information can be used instead of calculating the derivatives in each layer to produce similar learning, which reduces the computational cost, time and reciprocal connection dependency. However, they ignore the symmetric weights in the weight updating process unlike backpropagation. This might hamper the learning process while using dissimilarly capable inputs. Furthermore, in the feedback alignment methods each neuron requires feedback weights for each error in the network [45]. Therefore, they typically require a significant number of computation and data movement that compromises the locality of the algorithm [46]. Research also shows that these techniques perform poorly if the network becomes larger as the complexity and difficulty increases [45].

III. Methodology

A. Fusion and classification

We employed Xception and InceptionV3 pre-trained on ImageNet to extract visual features from the images. We also employed VGG16 pre-trained on the Places 365 dataset for scenic level feature extraction. To verify the impact of this feature set, we tested the classification in two phases: including the scenic oriented features, and excluding them. We used Python's "`keras`" library on google colab platform to extract the visual features.

For metadata, we took a text processing approach and exploited the user tags, the description and the title of the images. We preprocessed them by removing the texts that were only symbols and non-English terms and then enumerated the missing values with a fixed constant. The user tags of an image were provided as multiple words, which were joined together to form a single string for each image. Then we employed the BoW approach for the text feature extraction. We also eliminated the English stop words during the BoW process and extracted both uni-gram and bi-gram features. The bi-gram features capture the local ordering information between words. For example, a uni-gram such as "`heavy`" might not contain any flooding information, but a bi-gram such as "`heavy rain`" might. Finally, we calculated the term frequency-inverse document frequency (tf-idf) [47] of each feature. We used Python's "`scikit-learn`" library for text feature extraction. More specifically the functions "`CountVectorizer`" and "`TfidfTransformer`" were used to extract the BoW and tf-idf respectively.

We performed fusion of six modes, three visual modes: i) Xception, ii) InceptionV3 and iii) VGG16 and three textual modes: i) Tags, ii) Title, and iii) Description using both feature-level and decision fusion methods. To evaluate how each mode contribute in the classification, we included one mode at a time in the fusion process.

We employed a supervised Chi-squared method to select best features from each mode. Although, previous research shows that the feature reduction method can impact the classification accuracy [6], the main objective here was to make sure that each mode has same number of features for fair contribution in the fusion process. For example, VGG16 extracts 512 dimensional features, which is the minimum among all modes. Therefore, we selected 512 best features from every mode while we perform fusion including scenic features. For the other experiment, we selected 2048 features from each mode because that is the minimum number of features among all modes used in this experiment.

Using the described modes separately, we performed the image classification to get the posterior probabilities and averaged them for decision fusion. For feature-level fusion, we concatenated the features from each mode. We employed a neural network with our proposed learning method, DBP, for the classification task. We also tested different parameters such as the batch sizes, learning rates and total number of epochs for the neural network training. The batch size is the number of training examples taken at one forward pass, and epoch is the total number of iterations. The learning rate is amount that the weights are updated in every backward pass during the training process.

B. Direct Backpropagation

As an alternate to the backpropagation technique, we proposed a learning method that employs the cost function from the output layer for each layer's update, similar to DFA. This technique exploits the advantages of DFA and removes the reciprocal dependency of layers in the backward pass. However, unlike DFA, this technique uses the symmetrical weights in the gradient calculation process instead of a random fixed feedback. We call this technique the Direct Backpropagation method (DBP).

$$\delta a_2 = (W_3^T e) \odot f'(a_2), \delta a_1 = (W_2^T \delta a_2) \odot f'(a_1) \quad (1)$$

$$\delta a_2 = (B_2 e) \odot f'(a_2), \delta a_1 = (B_1 \delta a_2) \odot f'(a_1) \quad (2)$$

$$\delta a_2 = (B_2 e) \odot f'(a_2), \delta a_1 = (B_1 e) \odot f'(a_1) \quad (3)$$

$$\delta a_2 = (W_2 e) \odot f'(a_2), \delta a_1 = (B_1 e) \odot f'(a_1) \quad (4)$$

$$\delta a_2 = (W_3^T e) \odot f'(a_2), \delta a_1 = (W_2^T e) \odot f'(a_1) \quad (5)$$

Equation (1), (2), (3) and (4) show the hidden layer update direction for BP, FA, DFA and IFA respectively. Here, considering a three layered network, δa_2 and δa_1 are the second and first layer gradients respectively, $f'()$ is the derivative of the

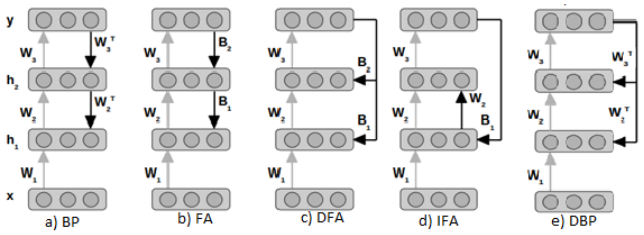

Fig. 1. Overview of different learning methods [43]

non-linearity, $\odot$ is an element-wise multiplication operator, B are random feedbacks, e is the gradient at the last layer or the cost function, W are the forward weights and W^T are the symmetric weights. Our proposed learning method can be presented by equation (5). Figure 1 shows the overview of different error propagation and parameter update techniques.

The DBP technique sends the the cost function, e from the last layer back to every layer to calculate their gradients, and therefore, removes the dependency on previous layer in the learning process. That means, once the forward pass is complete and the cost function is calculated, the gradients of all layers can be calculated at the same time in a parallel processing environment and thereby, reduces the processing time and cost. For example, in a deep neural network consisting hundred layers using BP as learning method, δa_1 could be calculated only after calculating all ninety nine previous δa, because δa_1 is dependent on δa_2, δa_2 is dependent on δa_3 and so on. This initiates a waiting time for each gradient calculation in the backward pass. However, with DBP, this dependency does not exist because the layers do not require information from previous layer on the backward path to update their gradients.

Moreover, the DBP technique utilizes the transpose of forward weights (W^T) in the backward learning process similar to BP. Therefore, unlike DFA, this technique does not require generating fixed random weight matrix for each neuron. Utilisation of transpose weights in the backward pass also maintains the locality of the algorithm. The transpose of weights is also expected to be effective than random feedback while using inputs of dissimilar credibility such as the visual modes and textual modes of social media.

IV. Results and Discussion

A. Dataset and experimental setup

The Disaster Image Retrieval from Social Media (DIRSM) task of MediaEval 2017 provided the dataset. It contains 6,600 Flickr images extracted from the YFCC100M-Dataset categorized by human assessors into two categories, flood (1) and no-flood (0). The dataset was separated with a ratio of 80/20 into two sets. The development-set contains 5,280 images and the test set contains 1,320 images. The organizers currently restrict public access to this dataset. To train the network, We perform the experiments with a 10-fold cross validation and average the results. Here, we split the training set into ten sets and use nine of them for training and one set for validation each time. We made sure that the training sets and the validation sets are similarly balanced with both categories.

TABLE I
Training accuracy with variable learning rates and epochs

Learning rate	**Epochs**	**Time (s)**	**Highest training accuracy(%)**
3×10^{-4}	500	680	99.1
3×10^{-4}	1000	1240	**99.3**
3×10^{-6}	500	768	91.3
3×10^{-6}	1000	1538	93.1

We tested different batch sizes including 64, 256, 512, 1024 and 4752. Here, for parameter selection experiments, we use two visual modes: InceptionV3 and Xception with one textual mode: Tags. Figure 2 shows the training accuracies over 1000 epochs with learning rate of 3×10^{-4} (a) and 3×10^{-6} (b) for different batch sizes filled in between their standard deviations.

The results show that with learning rate of 3×10^{-4}, we get a very similar stable learning for 64 and 256 batch sizes. Although, training with 512 batch size struggles to produce high training accuracy in the first 200 epochs, it reaches the peak after 350 epochs and performs as 64 and 256 batch sizes from there. However, the lower batch sizes require longer time for training. Alternatively, with larger batch sizes the learning process gets hampered because the number of weight updates is reduced. We compared training with 500 epochs and the result shows very small improvement in the accuracy with a cost of doubled training time (Table I) while using 1000 epochs . With a lower learning rate of 3×10^{-6}, the learning improves for the larger batch sizes. However, they fail to reach as high accuracy as of 3×10^{-4}. This is due to the fact that the learning is slower with a lower learning rate and therefore, requires more training to achieve higher accuracy. Therefore, we select batch size of 512 with learning rate 3×10^{-4} and 500 epochs as the parameters for our following experiments.

All experiments of this research were performed on a laptop computer with 2.6 GHz i7-6600U processor and 16 GB of memory.

B. Feature-level fusion

We concatenated the features from different modes and used them as the input pattern for the neural network for classification. Table II compares different modes separately and their feature-level fusions in terms of validation accuracy and their standard deviations (SD) (σ). It also shows improvement significance of the highest achieved accuracy based on the t-test results compared to with others. Here "s" stands for significant ($p \leq 0.05$) and "ns" for not-significant ($p > 0.05$) differences.

In this phase, we did not include VGG16 features and therefore, the number of features per mode was 2048. Results show that Xception produced the highest accuracy compared to other modes individually. However, the fusion of the modes improved the accuracy gradually. With all six modes, the number of fusion combinations were too many to present.

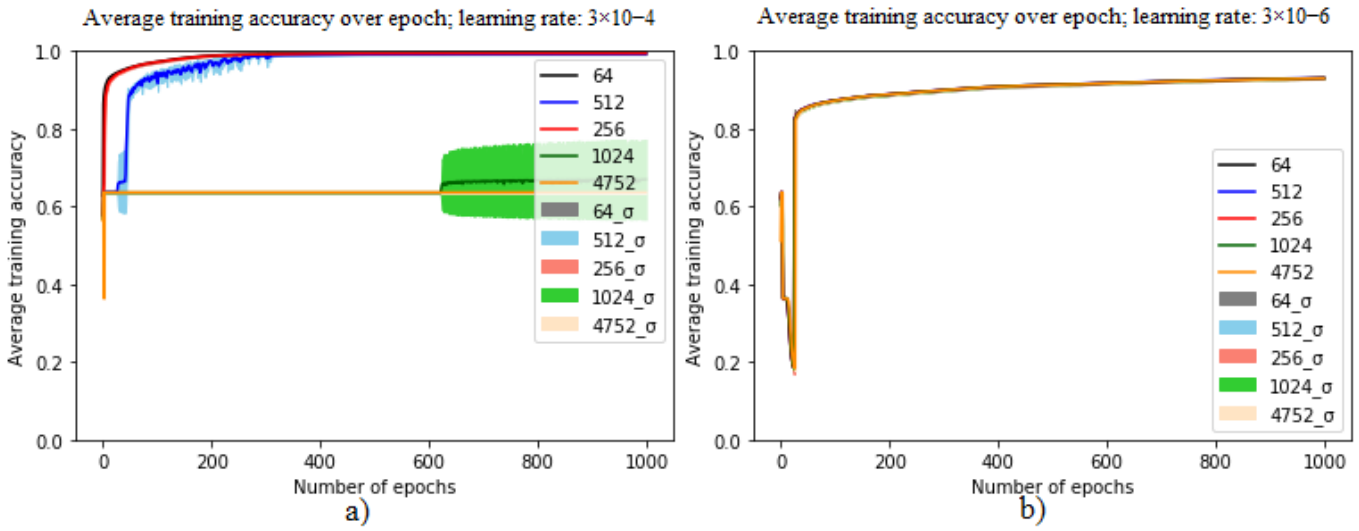

Fig. 2. Average training accuracy over number of epochs filled in between standard deviation using different batch sizes with learning rate 3×10^{-4} (a) and 3×10^{-4} (b)

TABLE II
CLASSIFICATION ACCURACY ON VALIDATION AND TEST SET USING DIFFERENT MODES SEPARATELY AND FUSED IN FEATURE-LEVEL

Modes(2048)	**Validation accuracy** (SD) (significance)	**Test accuracy**
Xception	90.7 (0.015) (s)	91.5
InceptionV3	89.6 (0.014) (s)	91.3
Tags	83.5 (0.015) (s)	81.9
Title	80.6 (0.018) (s)	78.0
Description	72.0 (0.012) (s)	68.8
InceptionV3, Xception	91.2 (0.013) (ns)	91.8
InceptionV3, Xception, Tags	91.2 (0.013) (ns)	92.1
InceptionV3, Xception, Tags, Title, Description	**91.3** (0.015)	**92.5**
InceptionV3, Xception, Tags, Title, Description	91.2 (0.011) (ns)	**92.5**

Therefore, we adapted a method where we included modes based on their classification accuracy in an descending order. For example, we combined Xception (accuracy = 90.7%) and InceptionV3 (accuracy=89.6%) first, then we included Tags (accuracy=83.5%) and then Title (accuracy=80.6%). Inclusion of Description did not improve the classification accuracy. In this phase, highest accuracy was achieved by fusing InceptionV3, Xception, Tags and Title.

Table III shows the results using 512 features from each mode. The result shows that the VGG16 features is capable of achieving good classification accuracy by itself. However, accuracies of other modes are lower compared their 2048 counterparts. This was due to the feature reduction on the original feature sets. The results show that the inclusion of textual metadata with visual features slightly improves the accuracy. However, inclusion of Title and Description improves the accuracy even further.

TABLE III
CLASSIFICATION ACCURACY ON VALIDATION AND TEST SET USING DIFFERENT MODES SEPARATELY AND FUSED IN FEATURE-LEVEL INCLUDING SCENIC LEVEL FEATURES

Modes(512)	**Validation accuracy** (SD) (significance)	**Test accuracy**
Xception	90.0 (0.018) (s)	90.7
inceptionV3	89.9 (0.014 (s)	90.6
VGG16	90.1 (0.012) (s)	91
Tags	83.0 (0.015) (s)	80.2
Title	79.4 (0.017) (s)	75.2
Description	71.4 (0.010) (s)	67.6
VGG16, Xception	90.9 (0.013) (ns)	90.6
VGG16, Xception, InceptionV3	91.1 (0.010) (ns)	91.0
VGG16, Xception, InceptionV3, Tags	91.0 (0.010) (ns)	**91.4**
VGG16, Xception, InceptionV3, Tags, Title	**91.1** (0.011)	91.2
VGG16, Xception, InceptionV3, Tags, Title, Description	91.0 (0.011) (ns)	91.2

TABLE IV
CLASSIFICATION ACCURACY ON VALIDATION AND TEST SET USING DECISION FUSION

Modes(2048)	**Validation accuracy** (SD) (significance)	**Test accuracy**
InceptionV3, Xception	91.3 (0.015) (s)	92.1
InceptionV3, Xception, Tags	92.4 (0.012) (s)	**93.4**
InceptionV3, Xception, Tags, Title	92.8 (0.012) (ns)	93
InceptionV3, Xception, Tags, Title, Description	**92.8 (0.008)**	92.8

C. *Decision fusion*

In the second step, we tested the decision fusion method. The results are presented in Table IV and Table V. Here also we perform the classification in two phases, using 2048 features and 512 features. The highest classification accuracy

TABLE V
CLASSIFICATION ACCURACY ON VALIDATION AND TEST SET USING DECISION FUSION INCLUDING SCENIC FEATURES

Modes(512)	**Validation accuracy** (SD) (significance)	**Test accuracy**
VGG16, Xception	90.4 (0.012) (s)	91.1
VGG16, Xception, InceptionV3	91.9 (0.008) (s)	92
VGG16, Xception, InceptionV3, Tags	92.9 (0.005) (s)	93.1
VGG16, Xception, InceptionV3, Tags, Title	93.2 (0.008) (ns)	93
VGG16, Xception, InceptionV3, Tags, Title, Description	**93.4 (0.007)**	**93.2**

was achieved by fusing InceptionV3, Xception, Tags, Title and Description using 512 features.

Overall the results show that the decision fusions produced higher classification accuracy than their feature-level counterparts that indicates the competence of decision fusion method for fusing heterogeneous modalities such as the high dimensional visual feature matrix and metadata sparse matrix. This also verifies previous researches [14], [34]. However, they require increased computational and processing cost. The results also show that the fused modes can produce significant improvement in classification compared to the individual modes.

D. Learning of weights

The results also distinguished visual features as superior than metadata for flood detection in social media images in Figure 3. It shows the last layer weights distribution of each feature after training the neural network using DBP. Here, we present weights of total 10,240 features (2048 features from each mode) from Xception, InceptionV3, Tags, Title, and Description sequentially concatenated together. Table VI summarizes the average absolute weights of each modes that shows that the visual features have around ten times higher weights than metadata.

We also compared our results with other baseline learning methods. Figure 4 shows the training average accuracy distribution over 500 epochs for DFA, IFA, FA, BP and our proposed method, DBP. We used top performing modes, InceptionV3, Xception and Tags fused in feature-level for this experiment. Result shows that the DFA, IFA and FA have smoother learning curves compared to BP and DBP. This can be explained by the use of fixed random weight matrix instead of symmetric weights in the backward pass in those three methods. Alternatively, both BP and DBP utilize the symmetric weights in the learning process that might cause the fluctuations through the learning process. However, for DBP, the fluctuation problem does not exist in the final 100 epochs.

Table VII summarizes the test results achieved using different learning methods on the test-set. The results show that DBP has the capability to produce highest classification accuracy and MAP compared to other methods. The FA method also produces very similar result. However, this method has the

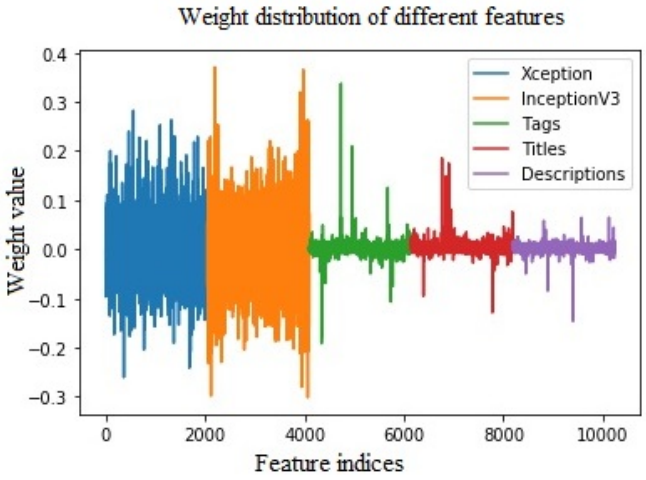

Fig. 3. Weights distribution of different features

TABLE VI
AVERAGE ABSOLUTE WEIGHTS OF EACH MODE

Modes	**Average weight**
IncetpionV3	**0.040**
Xception	0.029
Tags	0.005
Title	0.004
Description	0.004

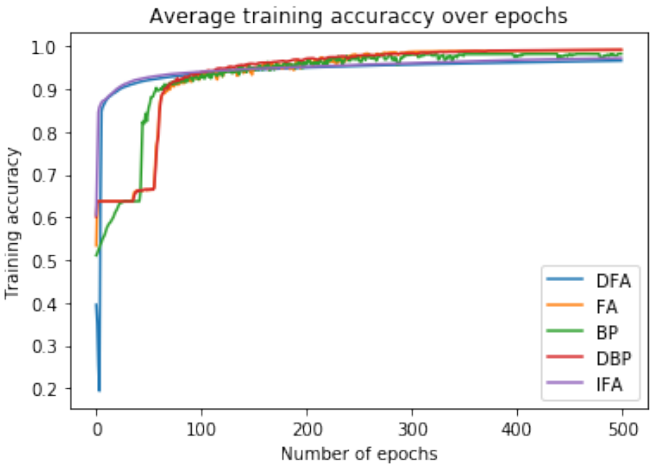

Fig. 4. (a) Training average accuracy using different learning methods over epochs, (b) zoomed to last 100 epochs

TABLE VII
RESULT COMPARISON WITH BASELINE LEARNING METHODS (ON TEST-SET)

Learning method	**Test accuracy (%)**	**MAP**	**Average training time per fold(s) (σ) (p)**
DBP	**92.2**	**96.1**	83 (0.72)
FA	92.1	96.0	85 (2.15) (s)
DFA	92.1	50.3	85 (3.46) (s)
IFA	91.9	51.6	85 (4.55) (ns)
BP	91.6	93.8	**82** (1.44) (s)

TABLE VIII
RESULT COMPARISON WITH TOP PERFORMING MEDIAEVAL 2017 APPROACHES (ON TEST-SET)

Approaches	MAP
DFKI	**97.4**
InceptionV3, Xception, Tags D 2048	**97.4**
InceptionV3, Xception, VGG16, Tags D 512	**97.4**
InceptionV3, Xception, Tags F 2048	96.1
MRLDCSE	92.5
InceptionV3, Xception, VGG16, Tags, Title F 512	91.0
ELEDIA@UTB	90.3
MultiBrasil	85.6
RU-DS	85.4

preceding layer dependency problem which is solved in DBP. IFA and DFA performed similarly in terms of test accuracy. However, both of them produced lower MAP. Upon investigation, we found that their generated posterior probability values were very marginal, which has hampered their performance in terms of MAP.

The result also showed that DBP takes significantly smaller average training time per fold compared to other methods except BP. This can be reduced further with parallel processing unlike in BP. Although the difference compared to IFA was not significant, the standard deviation of the training times was lowest with DBP.

E. External evaluation

We compared our results with top performing MediaEval 2017 fusion approaches in Table VIII using MAP across different cutoffs (k= 50, 100, 150, 240, 480) on test set. Therefore, MAP identifies the credibility of a method to retrieve top 480 relevant images. In the table, number of features are mentioned in the last part of the method names. A "D" and an "F" in the names indicate the decision fusions and feature-level fusions respectively.

Team DFKI has the highest MAP (97.4) in the comparison table, so does the fusion of InceptionV3, Xception and Tags fused in decision fusion manner and fusion of VGG16 with them. Team DFKI extracted 1000 dimensional visual features using X-ResNet pre-trained on DeepsentiBank and 200 dimensional metadata using the Word2Vec model on user tags only [19]. These two feature sets were concatenated to create 1200 dimensional feature for the classification. The idea of using only the user tags from metadata, which was only one-fifth of visual features in number, was effective here. Alternatively, team MRLDCSE took decision fusion approach and combined the posterior probabilities resulted from three support vector machines using features extracted by AlexNet pre-trained on ImageNet, the Places dataset, and metadata (user tags, Title and GPS information) [21]. They assigned equal weights to each mode manually during the combination process. However, the classification result using only metadata was very poor (MAP=18.23) compared to the only visual features (MAP=95.73). That indicated a clear advantage of visual feature over the metadata, and therefore, using equal weights on them degraded the overall performance. Team ELEDIA@UTB used an ensemble learner using two combined learner and a tuning learner to perform classification using visual features and took a text processing approach for metadata analysis [25]. They have also taken decision fusion approach and showed that the result was improved by 3% compared to visual only approach. Although they have employed a neural network for the metadata analysis, but exploited only the low-level visual features provided by the organizers. That describes their overall poor performance compared to top results. The rest of the teams did not produce competitive results.

Among our approaches, the decision fusion approaches performed better than the feature-level fusions. The highest MAP was achieved by fusing 2048 features from InceptionV3, Xception and Tags in a decision fusion manner. Furthermore, the decision fusion of 512 features from InceptionV3, VGG16, Xception and Tags produced a similar result. This indicates the impact of including scenic level features. Feature-level fusion of InceptionV3, Xception and Tags with 2048 features also produced high MAP. However, inclusion of 512 features of Tags and Title with InceptionV3, Xception and VGG16 could not produce as good of a result. This is because of the feature reduction and feature-level fusion of heterogeneous features. These approaches outperform most of the MediaEval 2017 approaches. These results also verify the weight values presented in Table VI that is the Tags and the visual features have higher capability of detect flood in social media images.

V. CONCLUSION

We have tested two fusion techniques, feature-level fusion and decision fusion to combine visual features and metadata for flood detection in Flickr Images. The results show that the fusion methods can improve the classification accuracy compared to their individual counterparts. Although, the decision fusion performs better than the feature-level fusion, it increases the computational cost. We have also investigated the impact of using scenic level features; however, this needs more investigation. The results distinguish the user tags as the most credible textual metadata. We have also investigated the learned weights for each feature using our proposed learning method, DBP, which has distinguished the visual features as more impcatful with higher values of weights. We have also compared our proposed learning method with other baseline methods. The results show that the DBP method can produce better results with lesser computational cost and processing time. Comparison of results with other fusion methods presented in MediaEval 2017 showed that our method is capable of outperforming most methods.

REFERENCES

[1] G. Cervone, E. Sava, Q. Huang, E. Scnebele, J. Harrison, and N. Waters, "Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 Boulder flood case study," International Journal of Remote Sensing, vol. I, pp. 100-124, 2016.

[2] C. Sarker, L. M. Alvarez, and A. Woodley, "Integrating recursive Bayesian estimation with support vector machine to map probability of flooding from multispectral Landsat data," in 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2016, pp. 1-8.

[3] R. I. Jony, A. Woodley, A. Raj, and D. Perrin, "Ensemble Classification Technique for Water Detection in Satellite Images," in 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1-8.

[4] A. Woodley, L.-X. Tang, S. Geva, R. Nayak, and T. Chappell, "Parallel K-Tree: A multicore, multinode solution to extreme clustering," Future Generation Computer Systems, vol. 99, pp. 333-345, 2019.

[5] R. S. Allison, J. M. Johnston, G. Craig, and S. Jennings, "Airborne optical and thermal remote sensing for wildfire detection and monitoring," Sensors, vol. 16, p. 1310, 2016.

[6] R. I. Jony, A. Woodley, and D. Perrin, "Flood Detection in Social Media Images using Visual Features and Metadata," in 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1-8.

[7] K. Ahmad, K. Pogorelov, M. Riegler, N. Conci, and P. Halvorsen, "Social media and satellites: Disaster event detection, linking and summarization," MULTIMEDIA TOOLS AND APPLICATIONS, vol. 78, pp. 2837-2875, 2019.

[8] J. Wang, M. Korayem, S. Blanco, and D. Crandall, "Tracking Natural Events through Social Media and Coputer Vision," in Proceedings of the ACM International Conference on Multimedia (MM), 2016, pp. 1097-1101.

[9] A. Woodley, T. Chappell, S. Geva, and R. Nayak, "Using web services to fuse remote sensing and multimedia data repositories," in Proceedings of the Australasian Computer Science Week Multiconference, 2017, pp. 1-8.

[10] E. K. Schnebele and G. Cervone, "Improving remote sensing flood assessment using volunteered geographical data," Natural Hazards and Earth System Sciences, vol. 13, pp. 669-677, 2013.

[11] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of Google Flu: traps in big data analysis," Science, vol. 343, pp. 1203-1205, 2014.

[12] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," IEEE transactions on pattern analysis and machine intelligence, vol. 41, pp. 423-443, 2018.

[13] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey," IEEE Access, vol. 7, pp. 63373-63394, 2019.

[14] 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), 2017, pp. 36-41.

[15] B. Bischke, P. Helber, C. Schulze, V. Srinivasan, A. Dengel, and D. Borth, "The Multimedia Satellite Task at MediaEval 2017."

[16] C. Sarker, L. Mejias, F. Maire, and A. Woodley, "Evaluation of the impact of image spatial resolution in designing a context-based fully convolution neural networks for flood mapping," in 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1-8.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 248-255.

[18] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 40, pp. 1452-1464, 2018.

[19] B. Bischke, P. Bhardwaj, A. Gautam, P. Helber, D. Borth, and A. Dengel, "Detection of flooding events in social multimedia and satellite imagery using deep neural networks," in Working Notes Proc. MediaEval Workshop, 2017, p. 2.

[20] C. Sarker, L. Mejias, F. Maire, and A. Woodley, "Flood mapping with convolutional neural networks using spatio-contextual pixel information," Remote Sensing, vol. 11, p. 2331, 2019.

[21] S. Ahmad, K. Ahmad, N. Ahmad, and N. Conci, "Convolutional Neural Networks for Disaster Images Retrieval," 2017.

[22] B. Bischke, D. Borth, C. Schulze, and A. Dengel, "Contextual enrichment of remote-sensed events with social media streams," in Proceedings of the 2016 ACM on Multimedia Conference, 2016, pp. 1077-1081.

[23] H. Tenkanen, E. Di Minin, V. Heikinheimo, A. Hausmann, M. Herbst, L. Kajala, et al., "Instagram, Flickr, or Twitter: Assessing the usability of social media data for visitor monitoring in protected areas," Scientific reports, vol. 7, pp. 1-11, 2017.

[24] M. A. Bashar, R. Nayak, N. Suzor, and B. Weir, "Misogynistic Tweet Detection: Modelling CNN with Small Datasets," in Australasian Conference on Data Mining, 2018, pp. 3-16.

[25] M. S. Dao, Q. N. M. Pham, and D. T. Dang Nguyen, "A domain-based late-fusion for disaster image retrieval from social media," 2017.

[26] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in Proceedings of the 13th annual ACM international conference on Multimedia, 2005, pp. 399-402.

[27] D. Kiela and L. Bottou, "Learning image embeddings using convolutional neural networks for improved multi-modal semantics," in Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP), 2014, pp. 36-45.

[28] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156-3164.

[29] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," arXiv preprint arXiv:1606.01847, 2016.

[30] J. Arevalo, T. Solorio, M. Montes-y-Gómez, and F. A. González, "Gated multimodal units for information fusion," arXiv preprint arXiv:1702.01992, 2017.

[31] E. Shutova, D. Kiela, and J. Maillard, "Black holes and white rabbits: Metaphor identification with visual features," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 160-170.

[32] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, et al., "Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention," IEEE Transactions on Multimedia, vol. 15, pp. 1553-1568, 2013.

[33] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, "Image and encoded text fusion for multi-modal classification," in 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1-7.

[34] B. Sun, L. Li, X. Wu, T. Zuo, Y. Chen, G. Zhou, et al., "Combining feature-level and decision-level fusion in a hierarchical classifier for emotion recognition in the wild," Journal on Multimodal User Interfaces, vol. 10, pp. 125-137, 2016.

[35] S. Planet and I. Iriondo, "Comparison between decision-level and feature-level fusion of acoustic and linguistic features for spontaneous emotion recognition," in 7th Iberian Conference on Information Systems and Technologies (CISTI 2012), 2012, pp. 1-6.

[36] S. Sun, "A survey of multi-view machine learning," Neural Computing and Applications, vol. 23, pp. 2031-2038, 2013.

[37] D. Wang, K. Mao, and G.-W. Ng, "Convolutional neural networks and multimodal fusion for text aided image classification," in 2017 20th International Conference on Information Fusion (Fusion), 2017, pp. 1-7.

[38] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in Advances in neural information processing systems, 2012, pp. 2222-2230.

[39] P. Wu, S. C. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao, "Online multimodal deep similarity learning with application to image retrieval," in Proceedings of the 21st ACM international conference on Multimedia, 2013, pp. 153-162.

[40] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency, "Efficient low-rank multimodal fusion with modality-specific factors," arXiv preprint arXiv:1806.00064, 2018.

[41] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," arXiv preprint arXiv:1805.11730, 2018.

[42] J. J. Hopfield, "Brain, neural networks, and computation," Reviews of modern physics, vol. 71, p. S431, 1999.

[43] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in Advances in neural information processing systems, 2016, pp. 1037-1045.

[44] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random feedback weights support learning in deep neural networks," arXiv preprint arXiv:1411.0247, 2014.

[45] S. Bartunov, A. Santoro, B. Richards, L. Marris, G. E. Hinton, and T. Lillicrap, "Assessing the scalability of biologically-motivated deep learning algorithms and architectures," in Advances in Neural Information Processing Systems, 2018, pp. 9368-9378.

[46] B. Crafton, A. Parihar, E. Gebhardt, and A. Raychowdhury, "Direct feedback alignment with sparse connections for local learning," Frontiers in neuroscience, vol. 13, p. 525, 2019

[47] J. Ramos, "Using tf-idf to determine word relevance in document queries," in Proceedings of the first instructional conference on machine learning, 2003, pp. 133-142.