Multi-Label Multi-Task Learning with Dynamic Task Weight Balancing

Tianyi Wang
School of Computing and Information Sciences
Florida International University
Miami, Florida
wtian002@cs.fiu.edu

Shu-Ching Chen
School of Computing and Information Sciences
Florida International University
Miami, Florida
chens@cs.fiu.edu

***Abstract*—Data collected from real-world environments often contain multiple objects, scenes, and activities. In comparison to single-label problems, where each data sample only defines one concept, multi-label problems allow the co-existence of multiple concepts. To exploit the rich semantic information in real-world data, multi-label classification has seen many applications in a variety of domains. The traditional approaches to multi-label problems tend to have the side effects of increased memory usage, slow model inference speed, and most importantly the under-utilization of the dependency across concepts. In this paper, we adopt multi-task learning to address these challenges. Multi-task learning treats the learning of each concept as a separate job, while at the same time leverages the shared representations among all tasks. We also propose a dynamic task balancing method to automatically adjust the task weight distribution by taking both sample-level and task-level learning complexities into consideration. Our framework is evaluated on a disaster video dataset and the performance is compared with several state-of-the-art multi-label and multi-task learning techniques. The results demonstrate the effectiveness and supremacy of our approach.**

***Keywords*-multi-label classification; multi-task learning; disaster video**

I. Introduction

In a real-world scenario, visual data often carry rich information describing a specific environment that contains multiple objects and their interactions [1]. The classic single-label classification deep neural networks are designed to detect the existence of a single object or action. Therefore, in order to model the rich semantic information in visual data, the deep neural networks should be adapted to model multiple objects. The specific task that aims to accomplish this goal is named multi-label classification. One important property that distinguishes multi-label classification from the common multi-class classification is that the labels in multi-label classification are not mutually exclusive. Recently, multi-label classification has attracted attention on a wide variety of domains [2][3].

Cost functions such as ranking loss, cross-entropy, and mean-squared error loss that are commonly used in single-label classification tasks cannot be directly applied to the multi-label classification problems. A widely adopted approach is to transform the problem into single-label classification. A classic method is one-vs-rest or one-vs-all. One-vs-rest splits the multi-label classification problem into several binary classification subproblems. Each subproblem has its classifier that is trained on one of the single labels. Then, the final prediction result is the ensemble of the output from all classifiers [4]. However, one major disadvantage of the native one-vs-rest method is the total disregard of inter-label dependency. It is well-studied that strong co-occurrence exists in a majority of the multi-label classification tasks [5][6]. A most prevalent case can be observed in natural disaster images, where victims and building debris frequently appear together. Many recent studies focus on learning the underlying correlation among labels so that the inter-label dependency could be retrieved [7][8]. Furthermore, in order to infer the joint label probability from the latent space, the final loss functions should be able to assess the optimal confidence thresholds for separating each label [9]. However, such approaches require precisely formulated modeling of the co-occurrence dependencies between labels, which can vary extensively among tasks and input sources [10]. Moreover, certain trade-off still needs to be made between the model complexity and the training time, since the pairwise correlation strategy adopted in these studies inevitably creates a large number of parameters [11].

Typical machine learning/deep learning models focus on solving a single task and have seen much success across many domains [12] [13]. However, the potential of exploiting multiple objectives simultaneously within a single model is quite attractive, since it not only reduces the memory consumption but also speeds up the inference process by making multiple inferences in a single pass. This type of approach is called multi-task learning. The most significant advantage of multi-task learning is its capability of utilizing the shared representations between related tasks. This is achieved by sharing network weights across multiple tasks. It helps the model to learn a better generalization of the problem and consequently reduce the risk of overfitting [14]. Multi-task learning has been applied in many fields, such as computer vision [15] [16], natural language processing [17] [18] and disaster management [19] [20] [21]. The common approach to optimize multiple tasks is to use the linear weighted sum on the loss of each task. Traditional methods of selecting the task weights are either using uniform weights or manually

978-1-7281-1054-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IRI49571.2020.00042

assigning the weights. If equal weights are used, it is highly likely for an easier task to deteriorate the overall model performance, since the very small loss will diminish the gradient of the aggregated loss during backpropagation. On the other hand, manually tuning the weights requires significant effort in searching for optimal weighting and is often time prohibited.

In this paper, we propose a novel multi-label multi-task attention network (MTMLAN) that utilizes the temporal and spatial information from the input data for video information retrieval tasks. Attention mechanism is applied to facilitate the training process by putting more weights on a specific segment of the input sequence. To address the challenge of the task weighting problem, we applied a novel dynamic weighting method that can automatically adjust the task weights based on both sample-level and task-level learning complexity. We evaluated our framework on a multi-label natural disaster video dataset, but it can also be expanded to almost any domain. The dynamic weighting method can be applied to all deep neural networks, which greatly expands its usability.

The key contributions of this paper are:

- A novel deep learning framework MTMLAN that utilizes multi-task learning to solve multi-label video information retrieval tasks.
- A novel dynamic task balancing method for multi-task learning problems that is based on the sample-level and task-level learning complexities.

The remainder of this paper is organized as follows. In Section II, the literature in multi-label and muti-task learning is briefly discussed. Section III presents a detailed description of the proposed MTMLAN framework and the dynamic task balancing method. Section IV illustrates the experimental results and discussions. Finally, in Section V we conclude the paper by discussing the main contributions and potential future research directions.

II. Related Work

A. Multi-label classification

Multi-label classification problems are more complicated than the traditional single-label problems due to the non-exclusive label in the input samples and their inherent generality nature [22]. Most network structures and loss functions in single-label classification problems cannot be directly applied to multi-label problems since they only function based on the binary concept assumption. Current state-of-the-art multi-label classification methods can be categorized into two groups: 1) problem transformation, and 2) algorithm adaptation.

Problem transformation methods aim to convert the multi-label problems into common single-label ones. Therefore, all existing approaches that are suited for single-label problems can be applied. One native method is the one-vs-all, where for each concept, a classifier is trained, with the samples from that concept treated as positive and the rest of the samples as negative [4]. The main drawback of this approach is the complete oversight of the correlation between concepts. To address the inter concept dependency relationship, label power set method creates multiple classifiers for each combination of the concepts [23]. However, as the concepts increase, the number of classifiers can grow exponentially, which leads to very few instances for each combination.

Algorithm adaptation methods tackle the problem by augmenting the existing single-label network structure to fit multi-label purposes. Wang *et al.* uses embedding layers to learn the joint label-image embedding which retains the co-occurrence label dependency and relevance [2]. In [24], the model optimization objective function is capable of approximating the multi-label neighborhood mutual information so that the input feature quality could be effectively measured. Such measurement is based on the mutual information it shares with a set of concepts. This feature selection technique helps the model learn the shared representation across labels.

B. Multi-task learning

Multi-task learning exploits the shared semantic information across tasks by training multiple tasks in the same model. Related tasks can complement each other so that the model can be more generalized. This leads to better training and inference efficiency, as well as stronger model performance. The most prevalent multi-task learning approaches can be categorized into 1) network architectures engineering, and 2) feature and task relation learning.

Existing deep neural network architectures can be tweaked to handle multi-task learning problems. The cross-stitch network [25] uses the cross-stitch units to combine multiple networks where each one of them is trained for a specific task. This helps the model to learn the optimal combination of shared and task-specific features. UberNet [26] builds the task-shared layer using a pyramid structure based on the VGG-NET [27]. It feeds a series of down-sampled images of different resolutions into the task-shared layer, which is constructed on top of the task-specific layers.

Exploiting the underlying relationship between feature and tasks can facilitate information sharing in multi-task learning. Lu *et al.* uses a dynamic branching approach to automatically construct a tree-like network structure. It places certain concepts in the same branch by considering the task correlation and complexity [28]. Also, the use of weight uncertainty [29] models the task-dependent homoscedastic uncertainty to weigh different tasks. In another work by Chen *et al.*, the gradient norms of each task-specific layer are used to dynamically change the learning progress [30]. Other measurements can also be used to balance the task weights. Dynamic task prioritization [31] uses the key performance indicator, such as accuracy or

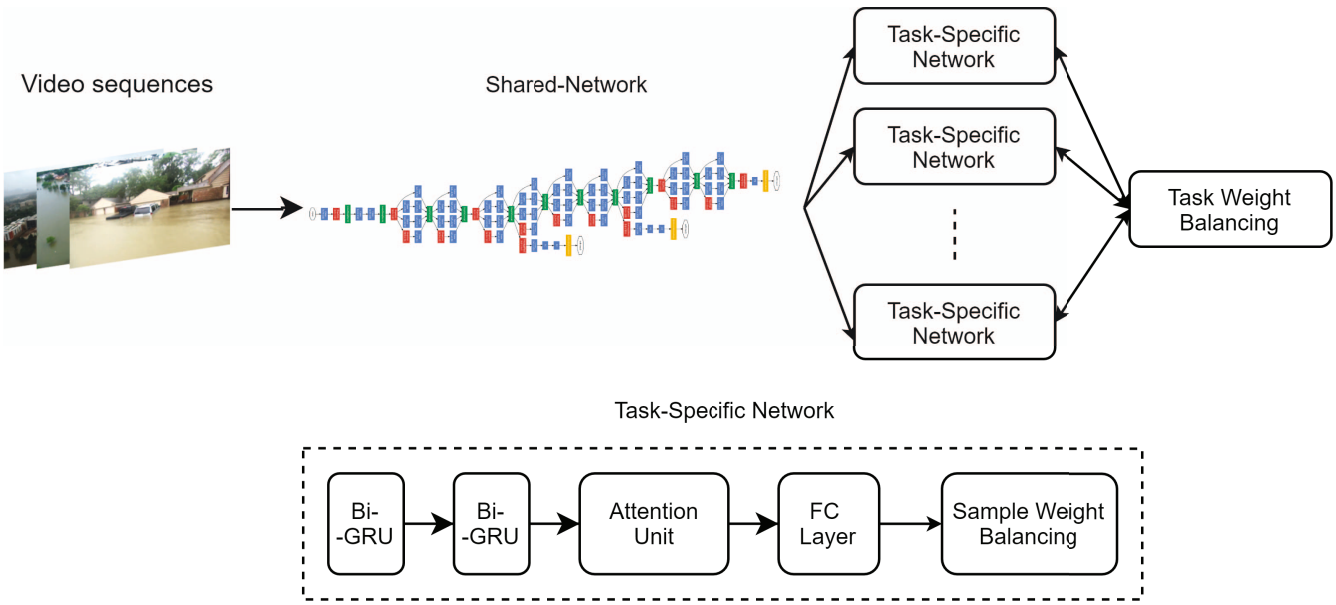


Figure 1: An overview of the proposed framework. The lower part of the diagram shows the anatomy of the task-specific network

average precision, as the learning progress signal to adjust the task weight distribution.

III. Methodology

In this section, the proposed multi-task multi-label attention network (MTMLAN) will be introduced. While this architecture could be applied to any deep neural network (DNNs) and problem domain, we use recurrent neural networks (RNNs) based DNN for the purpose of demonstration.

A. Architecture Design

The main architecture of MTMLAN follows the hard parameter sharing schema [32]. It consists of two main components: the shared network and the task-specific network. Figure 1 shows the overview of our proposed framework. The shared network learns the shared feature representation of all tasks, which can greatly reduce the risk of overfitting. In this paper, we construct the shared network based on the Inception V3 [33] model. The network consists of multiple small convolutional filters (3×3) and a batch normalized fully connected layer of the auxiliary classifier. More specifically, the original Inception V3 is truncated after the last average pooling layer to generate the spatial features.

The outputs of the shared network are then fed into each of the task-specific networks. The task-specific network contains two bidirectional Gated Recurrent Unit (BiGRU) and an attention module. The BiGRUs extract the temporal information from the sequential video frames. The attention module enables task-specific networks to learn task-specific features. In other work, it functions as a feature selection mechanism by helping the model focus on the most relevant part of the input sequence. In this paper, the attention module is implemented based on self-attention [34], a special variant of the attention mechanism. While in regular attention, the model looks at multiple sequence inputs at adjacent time steps to determine the weight to put on each location [35], self-attention helps the model to focus at different locations of the current input sequence to get a more in-depth representation of the subject matter.

We denote the input sequence as I and the number of features in I as n, then the input vector can be described as:

$$I = (\omega_1, \omega_2, ..., \omega_n) \tag{1}$$

where I is the output of the shared network and ω_i is the i_{th} feature for an n-dimensional input vector. Then, the BiGRUs takes the input sequence I and generates the hidden state h_m for feature m:

$$\overrightarrow{h_m} = \overrightarrow{GRU}(\omega_m, \overrightarrow{h_{m-1}}) \tag{2}$$

$$\overleftarrow{h_m} = \overleftarrow{GRU}(\omega_m, \overleftarrow{h_{m-1}}) \tag{3}$$

The vectorized hidden state H is a constituent of each feature-level hidden state h_i, which is the concatenation of the feature-level hidden states from the two unidirectional GRUs:

$$H = (h_1, h_2, ...h_n) \tag{4}$$

Then, the attention weight matrix M is calculated by using the hidden state vector H:

$$M = softmax(W_{s2}tanh(W_{s1}H)) \tag{5}$$

where W_{s1} and W_{s2} are two trainable weight matrices, $tanh()$ represents the hyperbolic tangent function. The softmax function ensures the sum of the weight matrix to be 1.

The attention weight matrix helps the model to focus on a specific location on the input sequence by assigning corresponding weights to each feature. To get the final weighted output A, we apply the attention weight matrix to the hidden state vector. Here the matrix multiplication operation is used:

$$A = MH \tag{6}$$

The weighted output is then fed into the last step of the task-specific network, which is the final fully connected layer.

B. Dynamic task balancing

Multi-task learning requires carefully balancing the training process between each task. The proposed dynamic task balancing method is comprised of two components: sample-level dynamic balancing and task-level dynamic balancing

1) Sample-level dynamic balancing: The traditional solution for the class imbalance problem is to assign a penalty factor to the majority class in the loss function. While effective, this method only considers the problem on a class level. The truth is, sample difficulty also has a substantial impact on the learning process. For instance, the cross-entropy (CE) loss function for binary classification tasks can be described as:

$$CE(p_k) = -\log(p_k) \tag{7}$$

where

$$CE(p_k) = \begin{cases} p, & if\ y-1 \\ 1-p, & otherwise \end{cases} \tag{8}$$

where $y \in \{-1, 1\}$ is the ground-truth label, $p \in [0, 1]$ represents the probability that the target class has label $y = 1$. Based on [36], we define the sample-level loss function $SL()$ as:

$$SL(p_k) = -(1-p_k)^{\beta}\log(p_k) \tag{9}$$

where α_t is β is the sample level focusing parameter.

As the sample is misclassified and p_k is small, $(1-p_k)^{\beta}$ is very close to 1. Therefore, the loss is not affected. In comparison, as p_k gradually turns to 1, the impact on loss will increase, which means the weight for correctly classified samples decreases. β controls the magnitude of how the weight of the easy samples decreases. As a result, the sample-level dynamic balancing method effectively helps the model to adjust the resources to difficult samples.

2) Task-level dynamic balancing: One of the most prominent issues with multi-task learning is to find suitable weights for each task so the weighted linear sum of all losses could be optimized. Inspired by [30], we propose a novel task-level dynamic balancing (TDB) method that is capable of handling the task imbalance problem. TDB uses the loss ratio between tasks as the metric to measure the task imbalances. The weight gradient from the first layer of the task-specific network is used to evaluate the current learning magnitudes. Therefore, the goal of the task-level loss function $TL(t)$ is to minimize the difference between the weighted gradient of each task and the average gradient weighted by the training rate. The task-level losses $TL(t)$ at training step t is defined as:

$$TL(t) = \sum_j \frac{N}{n_j} \left| G_W^{(j)}(t) - \overline{G}_W(t) \times \left[r_j(t)^{\theta}\right] \right|_1 \tag{10}$$

where N is the total number of training instances, n_j is the number instances in task j. $\frac{N}{n_j}$ is the inverted class/task distribution, which serves as a penalty term to suppress the majority class/task. W contains weight parameters from the last layer of the shared-network, α is a hyperparameter that governs how rapidly the training rate will be restored to the average scale, $G_W^{(j)}(t)$ represents the L_2 norm of the gradient of the weighted single-task loss $w_j(t)L_j(t)$ for task j with respect to the chosen weights W:

$$G_W^{(j)}(t) = \left\| \nabla_W w_j(t) L_j(t) \right\|_2 \tag{11}$$

$\overline{G}_W(t)$ defines the average gradient norm among all tasks T at time step t:

$$\overline{G}_W(t) = E_T\left[G_W^{(j)}(t)\right] \tag{12}$$

The relative inverse training rate of task j $r_j(t)$ is defined as:

$$r_j(t) = \frac{\hat{L_j}(t)}{E_T\left[\hat{L_j}(t)\right]} \tag{13}$$

where $\hat{L_j}(t)$ is the loss ratio for task j at time step t to time step 0:

$$\hat{L_j}(t) = \frac{L_j(t)}{L_j(0)} \tag{14}$$

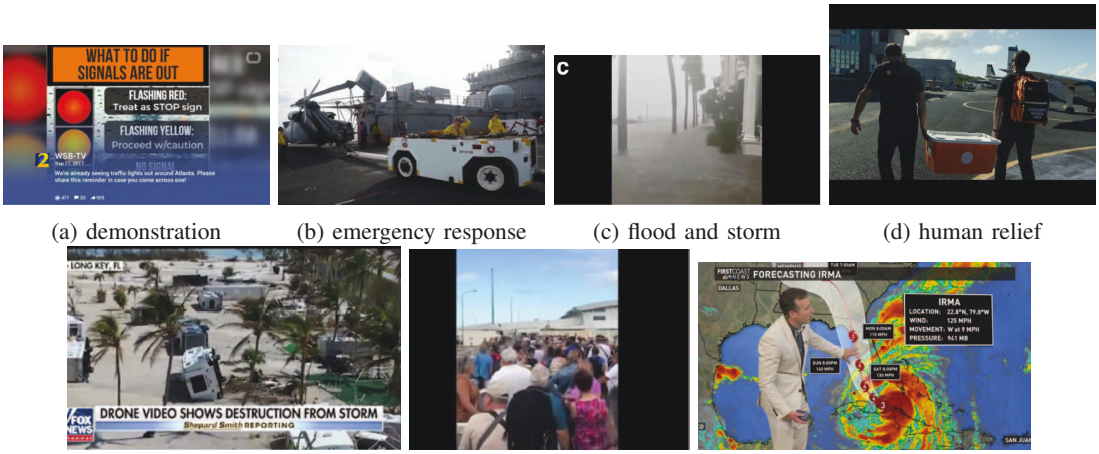
After upgrading the weight parameters in training step, the task losses are normalized so that the gradient will not be affected by the global training rate. The task weight for the next training set is then defined as:

$$w_j(t+1) = \lambda(t) w_j(t+1) \tag{15}$$

where

$$\lambda(t+1) = \frac{T}{\sum_j w_j(t+1)} \tag{16}$$

The steps to implement TDB for each training step can be described as: 1) Perform forward pass at the beginning of each training step, 2) extract the gradients of the first layer in each one of the task-specific networks G_W^j, and their corresponding L_2 norms are calculated, 3) calculates the average gradient $\overline{G}_W(t)$, 4) calculate the relative loss $\hat{t}$ for each task, 5) calculate the relative inverse training rates $r_j(t)$ for each task, 6) calculate the $\overline{G}_W(t) \times \left[r_j(t)^{\theta}\right]$ in equation 10, 7) calculate the gradient loss $TL(t)$, 8) update



(a) demonstration (b) emergency response (c) flood and storm (d) human relief

(e) damage (f) victim (g) briefing

Figure 2: Sample images of all concepts in the disaster video dataset

the task loss weights from $w_j(t) \rightarrow w_j(t+1)$, 9) update the model weights $W(t) \rightarrow W(t+1)$, 10) re-normalize the task loss weights $w_j(t+1)$

IV. Experiments

A. Dataset

In this work, we used a natural disaster video dataset [3] collected from YouTube. It contains 1,540 video clips and seven concepts (shown in Figure 2) that are related to 2017 hurricane Harvey and Irma. Following our previous work [37], each video clip is sub-sampled to 40 frames.

Table I: The statistical summary of the disaster video dataset

Concepts	Number of Instances	P/N Ratio
Demonstration	150	0.047
Emergency Response	338	0.105
Flood/Storm	971	0.301
Human Relief	273	0.085
Damage	371	0.115
Victim	311	0.096
Speak/Briefing/Interview	811	0.251
Total	3,225	

B. Experimental setup

The dataset is randomly split into 60% for training, 20% for validation, and 20% for testing. All hyperparameters are tuned on the validation set. The Inception V3 based shared-network is pre-trained on ImageNet [38] and the output of the last average pooling layer is used as the input for the task-specific networks. The proposed sample-level balancing loss function is used for each task-specific network, and the task-level balancing loss function is used on the final aggregated loss. Based on our empirical study, setting the hyperparameter α in the task-level loss function to 1 returns the best results. During the training, a batch size of 20 is used for the input. The learning rate is set to 0.001 and Adam [39] is used as the optimizer during the training. We report the results in Micro Averaged F-measure (MicroF1), Hamming Loss (HL), and Mean Average Precision (MAP).

C. Experimental Results

To demonstrate the effectiveness of our approach, several baseline methods are also tested on the disaster video dataset: 1) A common multi-label classification model (CMLC). It has similar network structure as the proposed MTMLAN before the task-specific networks. The task-specific networks are replaced with a single 2 layer Bidirectional GRU in this baseline model. The sigmoid activation function is applied on the last fully connected layer and cross-entropy is used as the loss function, 2) A equal weight multi-task classification model (EWMTC). It has the same network structure as MTMLAN without the sample-level and task-level dynamic balancing mechanism. Therefore, the final loss is simply the equal weight linear sum of all task losses, 3) GradNorm [30] is applied on MTMLAN to replace the proposed sample-level and task-level dynamic balancing mechanism, 4) Weight Uncertainty (WU) [29] method. The sample-level and task-level dynamic balancing mechanism in MTMLAN are replaced by the homoscedastic uncertainty approach, 5) MTMLAN without sample-level dynamic balancing (MTMLAN w/o SL Balancing), 6) MTMLAN without task imbalance penalty term (MTMLAN w/o TIP).

Table III shows the detailed performance results of the baselines and the proposed MTMLAN method. The "weight balancing" column shows which type of task weight balancing the corresponding method applies. It can be seen from the table that the common multi-label classification (CMLC) method has the worst performance regarding all three metrics. In comparison, the equal weight multi-task classification (EWMTC) method has better performance.

Table III: Performance evaluation results on the disaster video dataset

Approach	Weight Balancing	MicroF1	HL	MAP
CMLC	N/A	0.7267	0.1277	0.6848
EWMTC	Equal task weight	0.8015	0.1129	0.7341
GradNorm	Task-level	0.8569	0.0788	0.7822
WU	Task-level	0.8441	0.0793	0.7463
MTMLAN w/o SL Balancing	Task-level	0.8740	0.0661	0.8233
MTMLAN w/o TIP	Sample-level & task-level	0.8889	0.0634	0.8245
MTMLAN	Sample-level & task-level	**0.9135**	**0.0512**	**0.8559**

This illustrates the effectiveness of multi-task learning in solving multi-label problems.

The results of the two state-of-the-art multi-task learning techniques, namely Weight Uncertainty (WU) and GradNorm, demonstrate further improved performance compared to the vanilla EWMTC approach, with GradNorm having a slight edge over WU. It should be noted that both methods only focus on optimizing task level weight balance. Next, we compare the performance of the 3 variants of the proposed MTMLAN method. It can be seen from the table that both one of them outperformed GradNorm and WU. When purposely excluding the sample-level balancing function or the task imbalance penalty term in the task-level balancing function, the model performance did suffer. This demonstrates the effectiveness of the two components.

Table II shows the detailed classification accuracy for each task/concept of the baselines and the proposed MTMLAN method. It can be seen from the table that the trend for task-level classification accuracy performance is quite consistent with the overall performance of each method. The common multi-label classification (CMLC) method shows the worst performance among all 7 tasks/concepts, especially on tasks/concepts that have fewer samples. From Table I it can be observed that the P/N ratio of flood/storm and speak/briefing/interview concepts are significantly higher than the rest of the concepts in the dataset. This partly explains the worse performance on the minority concepts. In comparison, the equal weight multi-task classification (EWMTC) method has better performance, which again proves that by learning the shared-representation across all tasks, the model could generalize better on the whole problem domain. The same outcome applies to GradNorm and Weight Uncertainty, which further improve the accuracy across all tasks/concepts. However, none of these approaches shows a noticeable improvement in narrowing the performance gap between the minority and majority tasks/concepts.

In contrast, components in MTMLAN such as the sample-level balancing function and the task imbalance penalty term force the model to allocate more resources on difficult samples and minority tasks/concepts while training. As a result, minority tasks/concepts observed much higher performance gain compared to their majority counterparts. For instance, when using the results of CMLC as a benchmark, the accuracy of demonstration, damage, and victim concepts have improved by 14.74%, 21.79% and 18.53% respectively. This is significantly higher than the improvements on majority concepts such as flood/storm and briefing, which account for 10.22% and 7.55%.

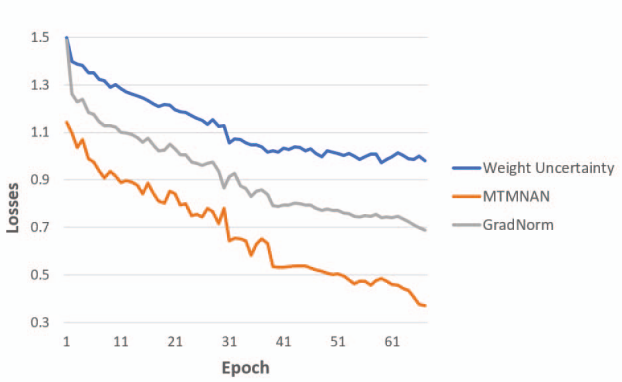


Figure 3: Training loss comparison among Weight Uncertainty, GradNorm, and the proposed MTMNAN methods

We further demonstrate the effectiveness of the proposed method in Figure 3, which shows the training loss history of MTMNAN against the other two state-of-the-art methods. It can be seen from the figure that MTMNAN constantly produces lower losses compares to the other two methods.

Table II: The per-concept accuracy results on the disaster video dataset

Approach	Demonstration	Emergency Response	Flood/Storm	Human Relief	Damage	Victim	Briefing
CMLC	0.8136	0.8066	0.8466	0.8123	0.7566	0.7452	0.8574
EWMTC	0.8249	0.8516	0.8779	0.8346	0.8010	0.7947	0.8552
GradNorm	0.8469	0.8711	0.9024	0.8753	0.8719	0.8540	0.8807
WU	0.8441	0.8597	0.9108	0.8776	0.8697	0.8600	0.8791
MTMLAN w/o SL Balancing	0.8740	0.9124	0.9137	0.9145	0.8913	0.8654	0.8985
MTMLAN w/o TIP	0.8948	0.9116	0.9194	0.9159	0.9083	0.8712	0.9001
MTMLAN	**0.9335**	**0.9487**	**0.9331**	**0.9410**	**0.9215**	**0.8833**	**0.9221**

V. Conclusion

This paper presents a novel multi-label multi-task deep learning framework for disaster video classification. The proposed MTMNAN model utilizes the shared-network to learn general information that can be shared across all tasks. On the other hand, the task-specific networks help the model learn patterns that are related to each task. The proposed dynamic task balancing approach automatically adjusts the training progress on both sample-level and task-level. The sample-level dynamic balancing function focuses on difficult instances by allocating more resources. At the same time, the task-level dynamic balancing mechanism adjusts weight distribution by attending to the training rate of each task. In addition, extra cautions are paid to the task imbalance problem by introducing the task penalty term to the task-level balancing function. In conclusion, we showed that the proposed MTMNAN is capable of achieving superior performance when compared to other state-of-the-art techniques. In the future, the proposed framework will be extended to accommodate multimodal data inputs. The interaction between different modalities could have a positive impact on the model performance and how to address the task balancing challenge across modality is worth investigating.

Acknowledgment

This research is partially supported by NSF OIA-1937019, NSF CNS-1952089, and NSF OIA-2029557.

References

[1] S. Chen and R. L. Kashyap, "A spatio-temporal semantic model for multimedia database systems and multimedia information systems," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 4, pp. 607–622, 2001.

[2] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.

[3] S. Pouyanfar, T. Wang, and S.-C. Chen, "A multi-label multimodal deep learning framework for imbalanced data classification," in *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2019, pp. 199–204.

[4] R. Babbar and B. Schölkopf, "Dismec: Distributed sparse machines for extreme multi-label classification," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 721–729.

[5] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, pp. 195–200.

[6] T. Mensink, E. Gavves, and C. G. Snoek, "Costa: Co-occurrence statistics for zero-shot classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2441–2448.

[7] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Effective supervised discretization for classification based on correlation maximization," in *2011 IEEE International Conference on Information Reuse & Integration*. IEEE, 2011, pp. 390–395.

[8] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, "Learning deep latent space for multi-label classification," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3617–3625.

[10] R.-W. Zhao, J. Li, Y. Chen, J.-M. Liu, Y.-G. Jiang, and X. Xue, "Regional gating neural networks for multi-label image classification." in *BMVC*, 2016, pp. 1–12.

[11] W. Weng, Y. Lin, S. Wu, Y. Li, and Y. Kang, "Multi-label learning based on label-specific features and local pairwise label correlation," *Neurocomputing*, vol. 273, pp. 385–394, 2018.

[12] Y. Yan, M. Chen, M. Shyu, and S. Chen, "Deep learning for imbalanced multimedia data classification," in *IEEE International Symposium on Multimedia, Miami, FL, USA, December 14-16, 2015*. IEEE Computer Society, pp. 483–488.

[13] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. E. P. Reyes, M. Shyu, S. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Survey.*, vol. 51, no. 5, pp. 92:1–92:36, 2019.

[14] D. Hernández-Lobato and J. M. Hernández-Lobato, "Learning feature selection dependencies in multi-task learning," in *Advances in Neural Information Processing Systems*, 2013, pp. 746–754.

[15] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald, "3d instance segmentation via multi-task metric learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9256–9266.

[16] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1480–1484.

[17] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018*.

[18] V. Sanh, T. Wolf, and S. Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6949–6956.

[19] L. Zheng, C. Shen, L. Tang, C. Zeng, T. Li, S. Luis, and S. Chen, "Data mining meets the needs of disaster information management," *IEEE Trans. Hum. Mach. Syst.*, vol. 43, no. 5, pp. 451–464, 2013.

[20] R. Chandra, "Dynamic cyclone wind-intensity prediction using co-evolutionary multi-task learning," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 618–627.

[21] T. Li, N. Xie, C. Zeng, W. Zhou, L. Zheng, Y. Jiang, Y. Yang, H. Ha, W. Xue, Y. Huang, S. Chen, J. K. Navlakha, and S. S. Iyengar, "Data-driven techniques in disaster information management," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 1:1–1:45, 2017.

[22] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[23] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[24] Y. Lin, Q. Hu, J. Liu, J. Chen, and J. Duan, "Multi-label feature selection based on neighborhood mutual information," *Applied Soft Computing*, vol. 38, pp. 244–256, 2016.

[25] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[26] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6129–6138.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[28] A. Lu, Yongxi hand Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5334–5343.

[29] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[30] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International Conference on Machine Learning*, 2018, pp. 794–803.

[31] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, "Dynamic task prioritization for multitask learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 270–287.

[32] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.

[33] 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*, 2016, pp. 2818–2826.

[34] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, Texas, USA, November 1-4, 2016*. The Association for Computational Linguistics, pp. 551–561.

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[37] S. Pouyanfar, Y. Tao, H. Tian, S.-C. Chen, and M.-L. Shyu, "Multimodal deep learning based on multiple correspondence analysis for disaster management," *World Wide Web*, vol. 22, no. 5, pp. 1893–1911, 2019.

[38] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, pp. 248–255.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.