Deep Spatio-Temporal Representation Learning for Multi-Class Imbalanced Data Classification

Samira Pouyanfar, Shu-Ching Chen
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
Email: {spouy001,chens}@cs.fiu.edu

Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
Email: shyu@miami.edu

***Abstract*—Deep learning, particularly Convolutional Neural Networks (CNNs), has significantly improved visual data processing. In recent years, video classification has attracted significant attention in the multimedia and deep learning community. It is one of the most challenging tasks since both visual and temporal information should be processed effectively. Existing techniques either disregard temporal information between video sequences or generate very complex and computationally expensive models to integrate the spatio-temporal data. In addition, most deep learning techniques do not automatically consider the data imbalance problem. This paper presents an effective deep learning framework for imbalanced video classification by utilizing both spatial and temporal information. This framework includes a spatio-temporal synthetic oversampling to handle data with a skewed distribution, a pre-trained CNN model for spatial sequence feature extraction, followed by a residual bidirectional Long Short Term Memory (LSTM) to capture temporal knowledge in video datasets. Experimental results on two imbalanced video datasets demonstrate the superiority of the proposed framework compared to the state-of-the-art approaches.**

***Keywords*-Deep learning; spatio-temporal learning; multi-class imbalanced data; video classification; CNN; LSTM.**

I. Introduction

Nowadays, multimedia big data analytics is highly important with extensive applications including intelligence surveillance, social network, healthcare, security, and robotics [1], [2]. It provides unprecedented opportunities to many real-world problems and situations [3], [4]. Among them, video classification is one of the most challenging and cumbersome tasks in multimedia big data. The main challenges in video classification are threefold: (1) There are large variations between the frames throughout the whole video (for example, the existence of various objects and scenes in one video such as tree, building, human, and water in a disaster event), (2) There are a large number of frames needed to be processed for each video, (3) The video data is multimodal and spatio-temporal in nature. Due to all these challenges, video content analysis and classification is a complex and big data problem requiring accurate and efficient learning models.

Many real-world problems are characterized as time series (e.g., human activity recognition, stock prediction, and sentiment analysis), and it is critical to discover the temporal patterns in a time series problem [5]. Accordingly, video data consisting of sequences of image frames can be considered as a time series problem in which both static and motion information need to be extracted and analyzed [6], [7]. However, most existing video classification techniques either ignore temporal information or utilize very complex motion features [8], [9] to model the temporal features which are not very efficient in practice.

One of the main challenges faced by the multimedia community is the non-uniform distribution of real-world datasets [10]. This problem is known as “data imbalance problem”, in which some of the classes contain much fewer samples than the others. Examples of the imbalanced data problem include rare disease identification, fraud detection, and natural disaster recognition. It has been widely shown in the literature that techniques such as data resampling (oversampling and undersampling) can enhance the prediction results of rare classes, especially for the binary classification tasks (e.g., cancer detection). However, it is challenging to employ such techniques on a multi-class imbalanced task while maintaining the temporal information on the video.

With the advent of deep learning, new methodologies have been proposed to address the problem of large-scale video classification [11]. Specifically, Convolutional Neural Networks (CNNs) [12] and Recurrent Neural Networks (RNNs) [13] are employed for modeling static and temporal information. Different from conventional machine learning algorithms, deep neural networks map large-scale raw data directly to the class outputs by automatically generating a hierarchy of features and classification scores. In contrast to the complex handcrafted visual features (e.g., Gabor, Histograms of Oriented Gradients (HOG), and Scale-Invariant Feature Transform (SIFT)), deep learning provides a general-purpose learning procedure resulting in discriminative features and high-level data abstraction. Existing work in video classification mostly trains two separate models for spatial and temporal learning [14]. Thus, the relationship between frame-based static information and sequence-based temporal information may not be accurately detected.

Despite the great success of deep neural networks in visual data classification, there remain challenges and rooms for improvement. To address these challenges, this paper presents a new deep learning framework that effectively handles the multi-class data imbalance problem using a spatio-temporal synthetic oversampling method. It also extracts static and temporal information from videos and reduces the overall training process using transfer learning. Specifically, a pre-trained CNN model is utilized to extract static features from video sequences which are later given to the proposed residual bidirectional LSTM model for spatio-temporal feature analysis. Finally, these discriminative features are directly fed to fully connected layers for the final class generation.

The remainder of this paper is organized as follows. Section II provides a brief study of the related work. In Section IV, the proposed framework is described. Section V provides the experimental results on a large video dataset. Finally, the paper is summarized in Section VI.

II. Related Work

A. Imbalanced Data Classification

Imbalanced data classification techniques are mainly classified into data level and algorithmic approaches [15]. The first group handles the imbalanced datasets by modifying the data distribution to balance the classes in the training set before applying the machine learning algorithms. The techniques in this group either decrease the frequency of the majority class (undersampling) or increase the frequency of the minority class (oversampling) [16]. Although these techniques can address the data imbalance problem, they may discard potentially important information or increase the likelihood of overfitting. More advanced techniques such as Synthetic Minority Over-sampling Technique [17] are proposed to avoid overfitting and information loss. The solutions of the latter group are algorithmic techniques in which the classifiers are designed to naturally handle the imbalanced datasets [18], [19]. Ensemble techniques such as bagging and boosting can improve the performance of classification and overcome the overfitting problem [20].

Existing work on imbalanced data classification is mainly limited to binary classification since multi-class imbalanced data classification has more complicated relations between its classes. An intuitive strategy to handle multi-class imbalanced data is to apply decomposition methods to turn the problem into a set of binary classification problems [21]. However, this method needs careful combination strategies to reconstruct the original multi-class dataset. Different from the existing work, in this paper, a multi-class classification model is proposed using the resampling techniques without decomposing the problem into binary classification. Furthermore, resampling is done through both spatial and temporal information in the video data.

B. Spatio-Temporal Video Analysis

Video classification is challenging due to its multimodality and spatio-temporal nature. Traditional methods combined several modality representations to enhance the classification performance. Chen et al. [22] proposed a multimodal data mining framework for semantic event detection from sports videos. Despite the great capability of the framework, it still needs human efforts for temporal analysis and also uses handcrafted features. In computer vision, several techniques have been proposed to detect motion and temporal information from videos. Among them, optical flow [8] and iDT [9] are able to generate discriminative motion features from the data. However, using engineering techniques for temporal analysis is a computationally expensive task.

Deep learning has been applied greatly in recent years to overcome the challenges of traditional methods and generate general-purpose models for feature analysis, either static or temporal [23], [24], [25], [26]. Spatio-temporal deep learning techniques can be divided into two groups: 1) Those generating separate models for each modality and fusing the information in the final layers [14], and 2) Those designing a comprehensive model to handle spatio-temporal information and their connections in one single model [27]. The 3D convolutional neural networks (called C3D) [27] fall under the second category that inherently applies both pooling and convolutional layers in the 3D space. In that work, the third dimension is time. This network requires very large-scale datasets to converge and very powerful and parallel machines including GPUs with high memory to train the deep 3D networks.

LSTM was originally proposed in 1997 [28] which is a variant of RNNs. Deep LSTM networks have been widely utilized in different applications such as NLP, speech processing, and time-series that require long-term temporal information. Specifically, it is used for video classification tasks in recent few years [29], [30]. Deep residual networks (ResNet) [24] were originally proposed by Microsoft Research (MSR) for an image competition task (ILSVRC 2015). This idea was later applied to many different applications and also video classification tasks [31].

All the aforementioned methods employ complex and computationally intensive handcrafted features such as optical flow [8] or iDT [9] for video classification and usually fuse several models to capture the spatio-temporal information. Moreover, these techniques usually ignore the imbalanced distribution of real-world data and are only evaluated on very balanced datasets. However, in this paper, an effective and efficient deep learning framework that integrates spatial and temporal features in a single model is proposed which also handles the data with skewed distributions.

III. Background

A. LSTM

LSTM networks have internal memory cells which are able to learn the long-term dependencies of sequential frames. In addition, they overcome exploding gradients in the temporal domain (vanishing problem) by providing temporal shortcut paths. Due to the simple input concatenation and activation applied in RNNs, it can remember information for a short time. Different from RNNs, LSTMs have a more complex structure assisting them to remember information for a longer period of time. As shown in Figure 1(a), when a new information arrives, the input gate i_t, forget gate f_t, output gate o_t, and memory cell c_t in the LSTM cell handle the information overwriting by comparing it with the inner memory. LSTM gates are designed to control the forgetting, updating, and remembering processes and enable gradients to smoothly flow through time. As a result, only the information that is needed are selectively passed.

Let σ be the sigmoid non-linearity which squashes the inputs to a range between $[0, 1]$, and $tanh(x)$ be the hyperbolic tangent non-linearity which squashes its input x to a range between $[-1, 1]$. The LSTM parameter updates at time step t given inputs x_t, h_t, and c_t are defined as follows [26]:

$$\begin{aligned}
i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i);\\
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f);\\
c_t &= f_t.c_{t-1} + i_t.tanh(W_c[h_{t-1}, x_t] + b_c);\\
o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o);\\
h_t &= o_t.tanh(c_t).
\end{aligned}$$

where W_k and b_k refer to the weight and bias of $k = \{i, f, c, o\}$, respectively. In order to gradually learn the connections of input i_t, forget f_t, and output o_t gates, they are component-wise multiplied by the input, hidden output, and memory cell.

B. Bidirectional LSTM

The original LSTMs have one direction and predict the output based only on previous information. Hence, some information may be lost in a one-directional network. Similar to human trajectories, Bidirectional LSTMs (BiLSTMs) are continuous and consider both former and subsequent information. As a result, it can capture bidirectional global temporal information in video sequences. Figure 1(b) illustrates a BiLSTM in which the input set is defined as $x = \{x_0, x_1, ..., x_t, x_{t+1}\}$ and the output set as $y = \{y_0, y_1, ..., y_t, y_{t+1}\}$ and the hidden layer as $h = \{h_0, h_1, ..., h_t, h_{t+1}\}$. In the hidden layers, there are forward sequences → and backward sequences ←. The parameters of BiLSTM at time t can be defined as follows [32]:

$$\begin{aligned}
h^{\rightarrow} &= g(U_{h^{\rightarrow}} x_t + W_{h^{\rightarrow}} + b_{h^{\rightarrow}});\\
h^{\leftarrow} &= g(U_{h^{\leftarrow}} x_t + W_{h^{\leftarrow}} + b_{h^{\leftarrow}});\\
y_t &= g(V_{h^{\rightarrow}} h^{\rightarrow} + V_{h^{\leftarrow}} h^{\leftarrow} + b_y).
\end{aligned}$$

where g is an activation function such as ReLu ($g(a) = Max(0, a)$), U refers to the weight matrix from the input to the hidden layers, W is the weight from the hidden to the hidden layers, V denotes the weight from the hidden to the output layers, and b_s denotes the bias of $s = \{h^{\leftarrow}, h^{\leftarrow}, y\}$.

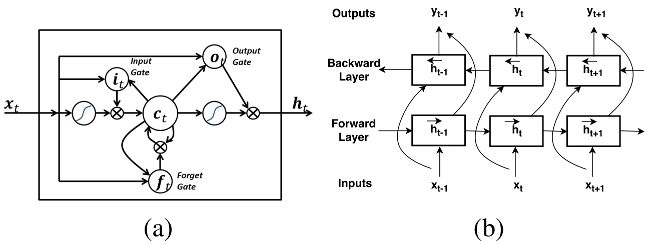


Figure 1. The architectures of (a) the LSTM cell and (b) unfold Bidirectional LSTM.

IV. Proposed Framework

The proposed framework is shown in Figure 2 which includes spatio-temporal synthetic oversampling, spatial, temporal, and prediction components. First, the video oversampling is employed to overcome the skewed distribution of the data, and then the static features of the video frames are extracted using the pre-trained CNNs. Thereafter, video sequences are generated and fed into the residual bidirectional LSTMs. Finally, the video classes are generated using the final fully connected layers.

A. Spatio-Temporal Synthetic Oversampling

Studies have shown that the use of sampling methods consisting the modification of the data distribution in an imbalanced dataset can help improve the classification performance. Thus, a new video oversampling method is proposed which includes two main components: random frame selection (temporal) and random augmentation (spatial). Suppose the multi-class training video dataset V includes N video samples and M classes ($V = \{v_{i,j} | i = 1, \cdots, N; j = 1, \cdots, M\}$, where $v_{i,j}$ refers to the i^{th} video sample belonging to the class j). The class set is $CL = \{cl_j | j = 1, \cdots, M\}$ where cl_j refers to the j^{th} class, that includes a different number of video samples nv_j. The maximum number of samples in a class set is δ and each video includes $frm_{i,j}$ frames.

Algorithm 1 illustrates the steps of the proposed spatio-temporal synthetic oversampling method which gets the video dataset V, the class list CL, δ, and α (sequence size) as the inputs and outputs the oversampled video dataset $\hat{V} = \{\hat{v}_{i,j,fr} | i = 1, \cdots, N; j = 1, \cdots, M; fr = 1, \cdots, freq_j\}$, where $\hat{v}_{i,j,fr}$ is the oversampled video related to the i^{th} video, j^{th} class, and fr^{th} frequency. The algorithm also generates the sequences of spatial features $Sequences = \{Seq_{i,j,fr} | i = 1, \cdots, N; j =$

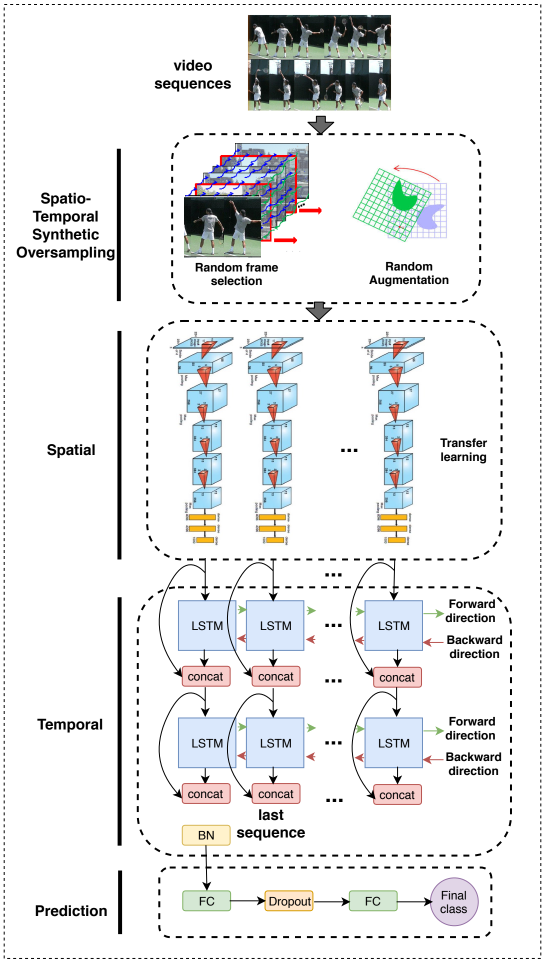


Figure 2. The Proposed framework for imbalanced video classification.

Algorithm 1: The proposed spatio-temporal synthetic oversampling algorithm for an imbalanced video dataset

```
Data: Original training video dataset
      $V = \{v_{i,j} | i = 1, \cdots, N; j = 1, \cdots, M\}$, Class
      list $CL = \{cl_j | j = 1, \cdots, M\}$, Maximum
      number of video samples $\delta$, and sequence size $\alpha$
Result: Oversampled video dataset $\hat{V} = \{\hat{v}_{i,j,fr} | i =
        1, \cdots, N; j = 1, \cdots, M; fr = 1, \cdots, freq_j\}$
        and Sequences of spatial features
        $Sequences = \{Seq_{i,j,fr} | i = 1, \cdots, N; j =
        1, \cdots, M; fr = 1, \cdots, freq_j\}$
$\hat{V} \longleftarrow \{\}$, $Sequences \longleftarrow \{\}$
for each class $cl_j \in CL$ do
    $freq_j \longleftarrow \left\lceil \frac{\delta}{nv_j} \right\rceil$
    for each video $v_{i,j} \in V$ do
        $frm_{i,j} \longleftarrow \mathrm{GetFrames}(v_{i,j})$
        for $fr = 1$ to $freq_j$ do
            if $SizeOf(frames_{i,j}) > \alpha$ then
                $frm_{i,j} \longleftarrow \mathrm{RandDown}(frm_{i,j})$
            else
                $frm_{i,j} \longleftarrow \mathrm{UpSample}(frm_{i,j})$
            $Seq_{i,j,fr} \longleftarrow \{\}$
            $\hat{v}_{i,j,fr} \longleftarrow \{\}$
            for each $Img \in frames_{i,j}$ do
                $\hat{Img} \longleftarrow \mathrm{RandAug}(Img)$
                $\hat{v}_{i,j,fr} \longleftarrow \hat{v}_{i,j,fr} + \hat{Img}$
                $features \longleftarrow \mathrm{SpatialFeatures}(\hat{Img})$
                $Seq_{i,j,fr} \longleftarrow Seq_{i,j,fr} + features$
            $\hat{V} \longleftarrow \hat{V} + \hat{v}_{i,j,fr}$
            $Sequences \longleftarrow Sequences + Seq_{i,j,fr}$
```

$1, \cdots, M; fr = 1, \cdots, freq_j\}$ where $Seq_{i,j,fr}$ is the feature sequence related to the oversampled video $\hat{v}_{i,j,fr}$. First, the frequency of oversampling for each class cl_j is calculated as $freq_j \longleftarrow \left\lceil \frac{\delta}{nv_j} \right\rceil$, where $\lceil\rceil$ is the ceiling function. In other words, the lower the number of samples in each class is, the higher the number of oversampling frequency will be. For example, if the maximum number of samples in all classes is $\delta = 100$ and the number of videos in the class j is $nv_j = 20$, then $freq_j = 5$. Therefore, this video is oversampled five times. Next, for each video $v_{i,j}$, the function $GetFrames()$ generates all the frames $frm_{i,j}$ in the video $v_{i,j}$. Since different videos have different numbers of frames, we turn each video into α-frames sequences. So, for each frequency (e.g., $\{1, \cdots, 5\}$), we either randomly downsample the frames to α-frames using $RandDown(.)$ function or upsample it to α-frames using $UpSample(.)$ function. If the number of frames in a video is higher than the specified sequence size (α), $RandDown(.)$ will return a random rescaled list of frames by getting a number to skip between iterations ($skip = \frac{|frm_{i,j}|}{\alpha}$) and then generating a random number for each $skip$. For example, if $\alpha = 5$ and $|frm_{i,j}| = 25$, then $skip = 5$ and a random number between one to five is selected in each iteration to generate the new rescaled frames.

The random frame selection process leads to a temporally oversampled dataset which can generate synthetic video samples from the original dataset. Although different frames are selected from each video in every iteration, they are spatially similar to each other, which may cause overfitting during the training. This is one of the main disadvantages of the oversampling techniques for imbalanced data. To overcome this issue, we utilize augmentation techniques for image samples. Essentially, we propose a random augmentation method $RandAug(.)$ which applies various image transformation to each oversampled video using random parameters. In other words, a random uniform distribution is used to generate different parameters for image transformation. Specifically, the image transformation function includes random rotation, translation, shear, and brightness. Finally, the new augmented image ($\hat{Img}$) is added as the frames of the new oversampled video $\hat{v}_{i,j,fr}$ and static features are generated for the corresponding image using the pre-trained models (e.g., IncaptionV3). These frames generated for each video are stitched together as a sequence $Seq_{i,j,fr}$ to be easily used in the next layers for video temporal analysis. Using this technique, the deep features are extracted once for each frame and may be used several times through the training process. Therefore, there is no need to continuously pass the original images through the CNN every time the same frame is read. The spatio-temporal synthetic oversampling algorithm returns these sequences to be used as the input of the temporal deep learning model.

B. *CNN-Residual Bidirectional LSTM*

As shown in Figure 2, the proposed deep learning model includes spatial, temporal, and prediction components. The video input (original and oversampled ones) flows in the spatial dimension (vertical direction) and temporal dimension (horizontal direction) and the corresponding classes are detected in the last prediction layer. In the spatial component, as explained in the previous section, deep CNN features are extracted for every frame from every video using transfer learning and converted into the sequences of extracted features. Several research studies have shown the effectiveness of deep features compared to the traditional handcrafted features [25]. In addition, utilizing pre-trained models can significantly expedite the whole training process on the new dataset. Depending on the target dataset and its similarity to the source dataset, the pre-trained CNNs can be truncated in various layers.

In the temporal component, the CNN feature sequences are fed into the proposed residual bidirectional LSTM as the time series to preserve the continuous temporal information. Residual connections can overcome the gradient transmission by forwarding the information from the upper layers directly through the network using an "addition" operator [24]. This simple connection can significantly improve the training process since the lower information can transmit to the upper layer directly through a highway. The residual connection provides not only the temporal shortcut paths but also an additional spatial shortcut path for efficient training of deep LSTM networks. Therefore, it gives a flexibility to the LSTM cells to deal with the vanishing or exploding gradients. Different from original LSTM, residual LSTM adds a shortcut path to the output layer h_t instead of accumulating a highway path on an internal memory cell c_t. The shortcut can be the output of any lower layers, though the exact previous output of Bidirectional LSTM is used in this paper. Then the network parameters are updated as follows:

$$\begin{aligned} h_0 &= \sigma(W_0 x + b_0); \\ h_l &= \sigma(W_l h_{l-1} + b_l) + h_{l-1}; \\ y &= \sigma(W_y h_{L-1} + b_y) + h_{L-1}. \end{aligned}$$

where $l = \{1, 2, ..., L-1\}$ and L is the total number of residual layers. In this paper, we use two residual layers (i.e., $L = 2$).

The proposed framework can access and discover more information in advance due to its backward passes and also can avoid overfitting and vanishing gradients due to its residual connection. In this paper, a two-layer residual bidirectional LSTM is designed ($L = 2$), followed by a batch normalization which is connected to the last element from its previous layer. In the final temporal layer, only the last element of the output is selected and batch normalization is applied because it normalizes the input across a mini-batch and generates simpler feature representations in the hidden layers. Therefore, it overcomes gradient vanishing and prevents outliers at the test time. In addition, L2 regularization is utilized to generalize the model and to reduce overfitting to the training data. More specifically, each parameter of the objective function is penalized by its squared magnitude as follows:

$$E(W) = \frac{1}{2}\sum_{n=0}^{N-1}(t_n - y(x_n, W))^2 + \frac{\lambda}{2}\|W\|^2$$

where $E(W)$ is the objective function, t_n is the actual class value of the n^{th} instance in the training batch, N is the total number of instances, and y is the output based on input x_n and weight W. The last term is the L2 regularization term including a penalty weight of λ.

Dropout is also directly added to each bidirectional LSTM layer. Dropout is a regularization technique which randomly ignores some neurons during the training, and so their contribution to the activation is temporarily deactivated. As a result, we can significantly prevent overfitting. Finally, the prediction component includes two fully connected layers and a dropout in between, which generates the final classes.

V. Experiments

A. *Experimental Setup*

In this paper, the proposed framework is applied to two video datasets to evaluate its performance, namely, the disaster video dataset introduced in [33] and public UCF101 action recognition dataset [5]. The disaster dataset was collected during two significant hurricanes (Irma and Harvey) and is naturally imbalanced. It includes seven classes (demo, emergency response, flood/storm, human relief, damage, victim, and speak) and the number of instances of each class varies from 40 to 400. On the other hands, UCF101 with 101 action categories is selected, which is one of the most challenging datasets due to its diversity in terms of actions, views, background, camera motion, and so on. However, different from the existing work on this dataset, the training set is resampled to serve for imbalanced video classification. To do so, a random number between 10 to the maximum number of instances in each class is generated and then those numbers of samples are randomly selected from each class. This means that each class contains at least 10 samples but may not include all of its original samples for training. The goal is to show how the proposed model can enhance the multi-class classification on a large-scale dataset with skewed distributions. The first train/test split of this dataset suggested by the reference website is used in this experiment.

In the preprocessing step, we first extract all the frames form each video. Thereafter, we extract the features of every video frame through the last pooling layer of InceptionV3, resulting in a feature set with 2048 dimensions. These extracted features are later grouped into sequences. For the sake of simplicity and similar to the experiments in [34], α is selected as 40. In other words, we turn each video into a 40-frame sequence. For temporal analysis, a two-layer Residual Bidirectional LSTM with 1024-wide followed by a 1024 fully connected layer and 50% dropout is used. This relatively shallow network outperforms other deep stacked Residual Bidirectional LSTM models. We use Adam stochastic optimization with an aggressively small learning rate 0.000001 and L2 regularization with $\lambda = 0.0003$.

The evaluation metrics used in this paper include Accuracy, F1, and Weighted F1 to consider both imbalanced data and multi-class classification.

B. Results and Analysis

Tables I and II summarize the experimental evaluation with the comparison against the state-of-the-art models on the disaster dataset and imbalanced UCF101, respectively. The comparison models include: (1) a model based on the CNN features and a simple LSTM. Although this model utilizes the temporal information using LSTM cells, it does not include any oversampling to handle the data imbalance problem; (2) the same CNN-LSTM architecture as the previous baseline, but in this model, the class weighting is added to automatically assign higher weights to the minority classes in the learning process; (3) the same CNN-LSTM architecture which also includes the proposed video oversampling; and (4) the same CNN-LSTM architecture which includes both video oversampling and class weighting. Finally, the last two rows show the results of the proposed CNN-ResBiLSTM without and with class weighting, respectively.

As shown in Table I, in the first group, no video oversampling is applied and it is assumed that deep learning can automatically handle the imbalanced data. It can be seen that both accuracy and F1 measures are significantly improved with a simple class weighting. This shows when the data samples of some of the classes are limited, it is necessary to assign a higher weight to these classes so that the learning algorithm will not bias toward the majority ones. In the second group, similar experiments are conducted plus applying the proposed spatio-temporal synthetic oversampling. It can be inferred from this set of results that the accuracy is boosted using the video oversampling. More importantly, the F1 measure is significantly improved, which shows the importance of this sampling technique over the weighting approaches. It is worth mentioning that the combination of oversampling and class weighting can enhance the performance results on this dataset since its highly imbalanced. Finally, the proposed model (CNN-ResBiLSTM) together with the proposed video oversampling and class weighting further improves the results and reaches to 70% accuracy and weighted F1. Compared to the original CNN-LSTM, the proposed techniques can enhance the accuracy and F1 measure by more than 11% and 0.17, respectively.

Similar experiments are conducted on the UCF101 with imbalanced distributions to further show the ability of the proposed framework on a large dataset. The results are shown in Table II which includes three sets of results: CNN-LSTM with no video oversampling, CNN-LSTM with video oversampling, and the proposed model. Each set includes the results with and without class weighting. Similar to the disaster dataset, data oversampling can improve the performance regarding both accuracy and F1 measures in a multi-class classification task. More specifically, the accuracy and F1 metric are improved by 1.5% and 0.3, respectively. The results are further improved by the proposed CNN-ResBiLSTM, which shows the importance of bidirectional and residual connections in our learning model. Different from the disaster dataset, the results are decreased when video oversampling is combined with the class weighting technique. Based on our observations, more overfitting happens for this dataset, which is a common disadvantage of class weighting and oversampling techniques. It is also due to the fact that the disaster dataset is much more imbalanced than the UCF-101 and needs more balancing strategies.

To further illustrate the effectiveness of the proposed residual bidirectional LSTM, we conduct several experiments on the UCF101 dataset as shown in Figure 3 (a-b). The figure visualizes the loss and accuracy comparison of each model during the training process. The comparison models include: (1) a frame-based CNN and softmax for generating final classes. This model called "Spatial CNN" which only considers static features in single frames and ignores the temporal information between the frame sequences. We fine-tune InceptionV3 by freezing the top layers of the network and updating the weights in only the final layers. This simple model surprisingly generates a promising performance compared to the more complex models; (2) a model based on the CNN features and a simple LSTM; (3) a model by adding residual connections to the previous model; (4) a model with bidirectional connections; and finally, our proposed model (ResBiLSTM).

It can be inferred from the plots that the proposed method can converge faster than the other benchmarks and generate lower losses and higher accuracies in almost all the itera-

Table I
PERFORMANCE EVALUATION RESULTS ON DISASTER DATASET.

Approach	Acc	F1	Weighted F1
No video oversampling			
CNN-LSTM	0.589	0.339	0.526
CNN-LSTM+ class weighting	0.663	0.428	0.654
With video oversampling			
CNN-LSTM	0.671	0.456	0.662
CNN-LSTM+ class weighting	0.678	0.477	0.688
Proposed model			
CNN-ResBiLSTM	0.681	0.493	0.678
CNN-ResBiLSTM+ class weighting	**0.700**	**0.513**	**0.706**

Table II
PERFORMANCE EVALUATION RESULTS ON IMBALANCED UCF101.

Approach	Acc	F1	Weighted F1
No video oversampling			
CNN-LSTM	0.685	0.655	0.670
CNN-LSTM+ class weighting	0.680	0.660	0.670
With video oversampling			
CNN-LSTM	0.706	0.684	0.696
CNN-LSTM+ class weighting	0.690	0.669	0.679
Proposed model			
CNN-ResBiLSTM	**0.723**	**0.702**	**0.717**
CNN-ResBiLSTM+ class weighting	0.705	0.686	0.696

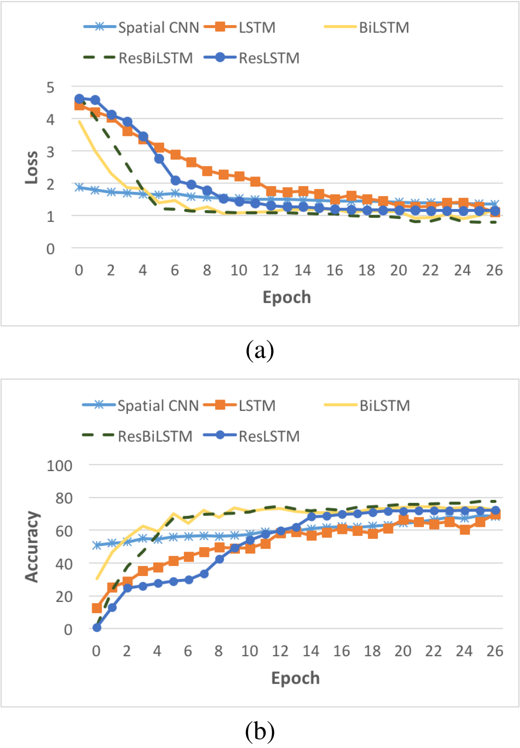
tions. The LSTM model has the slowest convergence while BiLSTM and ResLSTM can lessen this problem of LSTM. Finally, the proposed framework can learn forward and backward connections in each video sequence, leverage the temporal shortcut paths to expedite the training convergence, and reach to the higher performance faster.

VI. CONCLUSION

This paper presents a new spatio-temporal framework for large-scale and imbalanced video classification using deep learning. The framework introduces a new oversampling technique which generates synthetic videos to handle imbalanced data. Then, the spatial information is extracted from the video sequences using the pre-trained CNNs. Thereafter, these sequences are fed to the proposed two-layer residual bidirectional LSTM, and finally the video classes are predicted in the final fully connected layer. The experimental results demonstrate the ability of the proposed framework with respect to the prediction performance and efficiency.

ACKNOWLEDGMENT

This research is partially supported by NSF CNS-1461926.



(a)

(b)

Figure 3. Comparison of validation (a) loss and (b) accuracy on UCF101.

REFERENCES

[1] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyengar, "Multimedia big data analytics: A survey," *ACM Computing Surveys*, vol. 51, no. 1, pp. 10:1–10:34, 2018.

[2] H. Shahbazi, K. Jamshidi, A. H. Monadjemi, and H. Eslami, "Biologically inspired layered learning in humanoid robots," *Knowledge-Based Systems*, vol. 57, pp. 8–27, 2014.

[3] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, "Web media semantic concept retrieval via tag removal and model fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 61:1–61:22, 2013.

[4] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring, "Handling nominal features in anomaly intrusion detection problems," in *IEEE International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications*, 2005, pp. 55–62.

[5] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[6] H. Tian, H. C. Zheng, and S.-C. Chen, "Sequential deep learning for disaster-related video classification," in *The First IEEE International Conference on Multimedia Information Processing and Retrieval*, 2018, pp. 106–111.

[7] S.-C. Chen, M.-L. Shyu, and R. Kashyap, "Augmented transition network as a semantic model for video data," *International Journal of Networking and Information Systems*, vol. 3, no. 1, pp. 9–25, 2000.

[8] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[9] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[10] T. Meng and M.-L. Shyu, “Leveraging concept association network for multimedia rare concept mining and retrieval,” in *IEEE International Conference on Multimedia and Expo*, 2012, pp. 860–865.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[13] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, “Recurrent neural networks for emotion recognition in video,” in *ACM on International Conference on Multimodal Interaction*, 2015, pp. 467–474.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[15] C. Lemnaru and R. Potolea, “Imbalanced classification problems: systematic study, issues and best practices,” in *International Conference on Enterprise Information Systems*, 2011, pp. 35–50.

[16] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[18] C. Huang, Y. Li, C. Change Loy, and X. Tang, “Learning deep representation for imbalanced classification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5375–5384.

[19] X. Gao, Z. Chen, S. Tang, Y. Zhang, and J. Li, “Adaptive weighted imbalance learning with application to abnormal activity recognition,” *Neurocomputing*, vol. 173, pp. 1927–1935, 2016.

[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 4, pp. 463–484, 2012.

[21] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[22] M. Chen, S.-C. Chen, M.-L. Shyu, and C. Zhang, “Video event mining via multimodal content analysis and classification,” in *Multimedia Data Mining and Knowledge Discovery*, V. A. Petrushin and L. Khan, Eds. Springer-Verlag, 2007, ISBN:978-1-84628-436-6.

[23] M. Azimpourkivi, U. Topkara, and B. Carbunar, “A secure mobile authentication alternative to biometrics,” in *The 33rd Annual Computer Security Applications Conference*, 2017, pp. 28–41.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] S. Pouyanfar and S.-C. Chen, “Automatic video event detection for imbalance data using enhanced ensemble deep learning,” *International Journal of Semantic Computing*, vol. 11, no. 01, pp. 85–109, 2017.

[26] J. Kim, M. El-Khamy, and J. Lee, “Residual LSTM: Design of a deep recurrent architecture for distant speech recognition,” *CoRR*, vol. abs/1701.03360, 2017.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[28] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using LSTM,” in *International Conference on Machine Learning*, 2015, pp. 843–852.

[30] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[31] M. Marsden, K. McGuinness, S. Little, and N. E. O’Connor, “ResnetCrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification,” *CoRR*, vol. abs/1705.10698, 2017.

[32] Z. Yu, Y. Rennong, C. Guillaume, and G. Maoguo, “Deep residual bidir-LSTM for human activity recognition using wearable sensors,” *CoRR*, vol. abs/1708.08989, 2017.

[33] H. Tian, Y. Tao, S. Pouyanfar, S.-C. Chen, and M.-L. Shyu, “Multimodal deep representation learning for video classification,” *World Wide Web*, 2018.

[34] M. Harvey, “Five video classification methods implemented in Keras and TensorFlow,” https://blog.coast.ai/five-video-classification-methods-implemented-in-keras-and-tensorflow-99cad29cc0b5, 2017, accessed 10 Oct 2017.