Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification

Samira Pouyanfar[1], Yudong Tao[2], Anup Mohan[3], Haiman Tian[1], Ahmed S. Kaseb[4], Kent Gauen[5]
Ryan Dailey[5], Sarah Aghajanzadeh[5], Yung-Hsiang Lu[5], Shu-Ching Chen[1], Mei-Ling Shyu[2]

[1]School of Computing and Information Sciences, Florida International University, Miami, FL, USA
[2]Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA
[3]Intel Corporation, Santa Clara, CA, USA
[4]Faculty of Engineering, Cairo University, Giza, Egypt
[5]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
{spouy001, htian005, chens}@cs.fiu.edu, {yxt128, shyu}@miami.edu
anup.mohan@intel.com, akaseb@eng.cu.edu.eg, {gauenk, dailey1, saghajan, yunglu}@purdue.edu

Abstract

Many multimedia systems stream real-time visual data continuously for a wide variety of applications. These systems can produce vast amounts of data, but few studies take advantage of the versatile and real-time data. This paper presents a novel model based on the Convolutional Neural Networks (CNNs) to handle such imbalanced and heterogeneous data and successfully identifies the semantic concepts in these multimedia systems. The proposed model can discover the semantic concepts from the data with a skewed distribution using a dynamic sampling technique. The paper also presents a system that can retrieve real-time visual data from heterogeneous cameras, and the run-time environment allows the analysis programs to process the data from thousands of cameras simultaneously. The evaluation results in comparison with several state-of-the-art methods demonstrate the ability and effectiveness of the proposed model on visual data captured by public network cameras.

1 Introduction

Multimedia data, including visual, textual, and aural data, can be considered as the biggest big data due to its huge volume, high velocity, and variety [8]. Millions of network cameras (a type of surveillance cameras) have been deployed in city streets, tourist attractions, and many other locations. Network cameras are stationary and can capture and send real-time visual data (image or video) continuously over the networks without human effort. It is possible to use such rich data to observe long-term trends as well as to detect the anomaly. Figure 1 (a)-(d) show example images from network cameras, including roads, buildings, vehicles, and people. Figure 1 (e)-(i) show example images from several popular datasets. In contrast to the most commonly used datasets for classification, localization, object detection, and segmentation, network cameras contain many objects that occupy small portions of an image and are not necessarily at the center. This paper introduces the first version of the network camera image dataset.

Figure 1. Comparison of images from network cameras ((a)-(d)) and popular datasets ((e)-(i))

The data from many cameras are publicly available on the Internet. Even though the data are public, there is no easy way to find these cameras. Internet search usually finds vendors of cameras, not real-time data streams. Crowdsourcing [5] is one method to find these cameras. Different brands of cameras require different methods to retrieve data and crowdsourcing is not effective enough to handle the heterogeneity of the cameras. This paper presents a

systematic method to find network cameras. The cameras can be classified into three categories: Internet Protocol (IP) cameras that support HyperText Transfer Protocol (HTTP), non-IP cameras whose data is aggregated at web servers, and cameras using Real Time Messaging Protocol (RTMP). This system uses only publicly available data, and has been reviewed by the Institutional Review Board (IRB) and the legal counsel of the authors' university.

Most real-world data has a long tail distribution. In other words, some of the concepts are very scarce while others are abundant. This phenomenon is known as "imbalanced data problem" [12] which is widely seen in different applications such as medical, object classification, and surveillance systems. The problem is to classify the minority cases from the overwhelming majority cases correctly.

Despite the great success of deep learning algorithms in recent years, very few studies on deep learning consider classification for imbalanced data [7]. The existing deep neural networks such as CNNs can achieve very high performance using a balanced dataset (e.g., CIFAR, MNIST, Caltech, etc.) compared to the conventional classifiers. However, based on our empirical study, they perform worse in imbalanced datasets since they were not originally designed to address this problem. In addition, in current studies, few evaluation metrics have been utilized to accurately measure the performance of the deep learning models on the minority concepts.

These challenges motivate us to propose a new deep learning model to tackle the class imbalance problem in real-world data. This model modifies the existing CNNs to handle imbalanced data for multi-class classification in an effective manner. For this purpose, the proposed model dynamically modifies the samples of each class in each iteration based on the F1-score of that class in the reference set. We propose to integrate the scores of the F1-based model with the basic CNN model and utilize data augmentation and transfer learning (fine-tuning the pre-trained models) techniques to avoid overfitting toward the minority classes and to generalize the model. This approach will significantly improve the performance of the minority classes and maintain the performance of the majority ones.

The key contributions of this research study include: (1) It proposes a new deep learning model to handle the class imbalance problem in a multi-class classification problem. (2) It presents a systematic method to discover network cameras. (3) It introduces the first version of network camera image datasets which include thousands of images containing various real-world scenes.

2 Related Work

Visual data analysis is challenging because it is large-volume and heterogeneous in nature. Analyzing this massive amount of data is computationally expensive. Over the last decade, many studies have been conducted to manage and analyze visual data (images and videos) efficiently and effectively. Deep learning has shown significant advancement in the last few years. Inception [13], ResNet [4], and VGGNet [10] are examples of successful deep learning networks in image classification. All such models require training on large-scale datasets with enough data representations for each class.

Recent studies on the class imbalance problem can be mainly divided into two groups: data and algorithmic. The former manipulates the data distribution by either undersampling the majority classes or oversampling the minority ones to provide a more balanced dataset. However, oversampling may cause overfitting; while undersampling may lose some useful information. Synthetic Minority Oversampling Technique (SMOTE) [2] is proposed to handle overfitting problem by synthetically generating minority instances in feature-level rather than data-level using linear interpolation between nearest neighbors. On the other hand, the algorithmic techniques try to handle the data imbalance problem by improving the learning procedure [1]. However, these techniques usually increase the computational cost and training time to generalize the model.

The aforementioned studies do not address the class imbalance problem in CNNs. Lately, few studies integrated existing class imbalance solutions into the deep learning algorithms. Yan et al. [14] proposed a bootstrapping approach combined with CNNs to handle the minority classes in a binary classification task. A cost-sensitive learning was proposed for better feature representation learning from CNNs [6]. Different from existing techniques, an effective dynamic sampling in CNNs is proposed to enhance classification performance of both minority and majority classes.

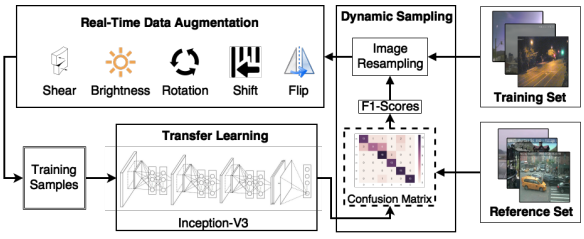

Figure 2. The proposed model.

3 The Proposed Deep Learning Model

The proposed model is depicted in Figure 2, which includes real-time data augmentation module, CNN transfer learning module, and dynamic sampling module. Real-time data augmentation module is used to generate the transformed images for each training batch, transfer learning

module is utilized to fine-tune the model, and dynamic sampling module is designed to automatically generate new samples based on the performance of the reference set.

3.1 Real-time Data Augmentation

Augmentation can improve the generalization and prevent overfitting while reducing the need for large-scale datasets. This process can be done either offline before training the model or real-time in each iteration of learning. In offline augmentation, we need to re-create the dataset before starting the training process. However, in real-time augmentation, we only transform a small batch of images that are required for each training iteration. In this step, we generate batches of image data via real-time data augmentation. This approach directly augments the input data to the model in the data space.

3.2 Transfer Learning in CNNs

In Transfer Learning (TL), given a pre-trained source domain $D_S = \{X, P(X)\}$ and a source task $T_S = \{Y, f(.)\}$, where X is the feature space, Y is the label space, and $P(X)$ is the marginal probability distribution (i.e., the probability distribution of the variables contained in X), the goal is to improve the learning of the target objective function $f_T(.)$ in target task T_T by transferring the knowledge from a source domain D_S to a target domain D_T, where either $D_S \neq D_T$ or $T_S \neq T_T$. In this paper, T_S is the ImageNet classification task, and T_T is our network camera classification task. Hence, the goal is to enhance the concept detection objective function $f_T(.)$ using the knowledge from ImageNet.

In this proposed work, the "InceptionV3" model M_S (originally pre-trained on ImageNet) is used to fine tune the model on our dataset in two levels. First, we train only the top layers that are randomly initialized and freeze all convolutional layers. Thereafter, only the top inception blocks are trained, and the other first layers are frozen. This is motivated by the fact that the early layers generate more generic features (such as color, edge, and shape) which can be leveraged in many tasks. However, later layers generate more specific features related to the original dataset.

3.3 Dynamic Sampling in CNNs

The training data used for transfer learning is critical to calibrate the parameters in the CNN model and those imbalanced classes might not be learned well. In this paper, the dynamic sampling mechanism is proposed to tackle this problem. Inspired by the experience that humans practice similar questions to avoid the same error happening again, we utilize the performance metric on the reference dataset to adjust the class distribution of the training samples, and thus indirectly effects the training process. Here, the F1-score metric is used, and the score is calculated based on the one-against-all assumption for each class.

Algorithm 1: Model Training Framework with F1-Based Dynamic Sampling

Data: Training Images X_{train}, Reference Images X_{ref}, Initial Model M_S, and Class List C
Result: Dynamic-Sampling-Based Model M^1

$M_0^1 \longleftarrow M_S$, $i \longleftarrow 1$, $N^* \longleftarrow \frac{|X_{\text{train}}|}{|C|}$
for *each class* $c_j \in C$ **do**
    $N_{i,j} \longleftarrow N^*$
while $\neg\text{IsFullyTrained}(M_{i-1}^1)$ **do**
    $X_i \longleftarrow \text{ImageSampling}(X_{\text{train}}, N_i)$
    $M_i^1 \longleftarrow \text{Train}(M_{i-1}^1, X_i)$
    $F1_i \longleftarrow \text{UpdateF1}(M_i^1, X_{\text{ref}})$
    for *each class* $c_j \in C$ **do**
        $N_{i+1,j} \longleftarrow \text{UpdateSampleSize}(F1_i, c_j)$
    $i \longleftarrow i + 1$
$M^1 \longleftarrow M_{i-1}^1$

In the proposed framework, the target model M^1 is initialized by M_S, trained by the set of images X_{train}, and the dynamic sampling is performed based on the set of images X_{ref}. All the classes are given in the list $C = \{c_j\}$. Algorithm 1 shows the model training framework with dynamic sampling, where $|X_{\text{train}}|$ is the size of the training dataset and $|C|$ is the number of classes. In iteration i, the training images X_i are sampled from X_{train} in the target domain and augmented as mentioned in Section 3.1. The number of the images of each class is determined by its F1-scores in the previous iteration $F1_{i-1}$. After the model is trained by the generated sample set, the updated model is used to predict the concept of each image in the reference dataset X_{ref}, where the images are completely different from those in the testing dataset (obtained from either a different camera or different time). The F1-scores of class c_j in iteration i, $f1_{i,j}$, are thus calculated.

$$f1_{i,j} = \frac{2 \cdot \text{Rec}_{i,j} \cdot \text{Pre}_{i,j}}{\text{Rec}_{i,j} + \text{Pre}_{i,j}} \tag{1}$$

where $\text{Pre}_{i,j}$ and $\text{Rec}_{i,j}$ are the precision and recall metrics of the class c_j in iteration i. Note that $F1_i = \{f1_{i,j}\}$ is the vector of the F1-scores of all the classes in iteration i. If a class has a higher F1-score, it can be better distinguished from the other classes in C. Hence, it becomes more important to improve the performance of the classes with lower F1-scores and thus more samples (images) from these classes will be selected in the next iteration; while the total number of images trained in each iteration remains the same. Eq. (2) defines the number of images of c_j of the next

iteration. The number of images in any class c_j is initialized to N^* which is the average number of samples in all classes, i.e., $f1_{0,j} = N^*$.

$$\text{UpdateSampleSize}(F1_i, c_j) = \frac{1 - f1_{i,j}}{\sum_{c_k \in C} (1 - f1_{i,k})} \times N^* \quad (2)$$

3.4 Model Fusion

For image inference in the testing stage, a single model might not work well for all the classes. The model trained with dynamic sampling tends to perform better on most of the imbalanced classes. Therefore, we propose to fuse two models, where one model M^1 performs better for the imbalanced classes and the other model M^0 performs better for the relatively balanced classes, to further improve the classification performance. The model M^0 can be obtained by any transfer learning without the sampling. The final inferred class can be decided by the results from both models, as shown in Eq. (3).

$$\mathcal{J} = \begin{cases} \mathcal{J}_0, & R_{\mathcal{J}_0} > Tr \quad \& \quad R_{\mathcal{J}_1} > Tr \\ \mathcal{J}_1, & R_{\mathcal{J}_0} < Tr \quad \& \quad R_{\mathcal{J}_1} < Tr \\ \mathcal{J}_{\underset{\theta \in \{0,1\}}{\text{argmax}} \left[M^\theta(X, c_{\mathcal{J}_\theta}) \right]}, & \text{otherwise} \end{cases} \quad (3)$$

where $\mathcal{J}_\theta = \underset{c_j \in C}{\text{argmax}}(M^\theta(X, c_j))$, $M^\theta(X, c_j)$ is the score of c_j calculated by model M^θ for the input image X, R_j is the ratio of the positive to negative samples (P/N ratio) of c_j, and Tr is the threshold of the P/N ratio to determine the choice of the model. Tr is determined by the model performance on the validation dataset.

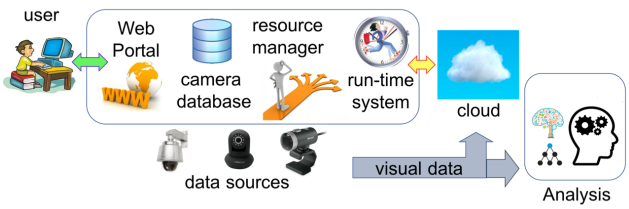

Figure 3. The main components of the proposed system

3.5 System Description

3.5.1 System Overview

To take advantage of the real-time visual data from thousands of network cameras, we have built an open-source system[1]. Figure 3 illustrates the main components of the system: (a) Web user interface. A user can find cameras at specific locations and analyze the data from these cameras. (b) Camera database. Currently, the database has more than 120,000 cameras deployed worldwide by governments, universities, research labs, etc. (c) Run-time environment. The system can retrieve real-time data and then analyze the data immediately. The run-time system adopts an event-driven programming interface. (d) Resource manager. The system uses the cloud (Amazon EC2 or Microsoft Azure) for retrieving and analyzing the data and also uses the cloud (Amazon S3) for storing the data. (e) Visual data analysis. This component analyzes the real-world image/video data which can be further utilized in information retrieval systems. To the best of our knowledge, this is the only system that can perform real-time analysis of data from thousands of network cameras.

[1]The source code will be made available at https://www.cam2project.net/

3.5.2 Discovering Heterogeneous Network Cameras

Although many network cameras provide data to the public, finding them is not always easy because of the wide variety of brands and models. The data from many network cameras are aggregated on web servers, and there are many different ways of organizing the data streams. To use the data from the network cameras, this system has a database of publicly available network cameras. The process for finding the network cameras depends on the types of the cameras. Some cameras have built-in web servers to distribute the data and each camera has a unique IP address. The data can be accessed by connecting directly to this address and using the HTTP GET requests. Different brands require different GET commands. For example, Axis cameras use /mjpg/video.mjpg but Foscam uses /video.cgi. Many organizations, such as Department of Transportation, deploy cameras and make the data available to the public. Retrieving data from these cameras requires sending the HTTP GET request to the web servers, not to the cameras directly. Some cameras employ RTMP stream and are rendered in the user's browser using Adobe Flash player. A program connects to the server distributing the video streams and monitors the network connections. The system inspects the handshake that sets up the video streams. Once the handshake request has been obtained, the system can connect and save the stream using rtmpdump [11]. This system has the same programming interface to analyze data from all the network cameras.

4 Experiments and Analysis

4.1 Dataset Description and Preprocessing

For this experiment, the research team runs the archiver program to retrieve an image from every active camera in

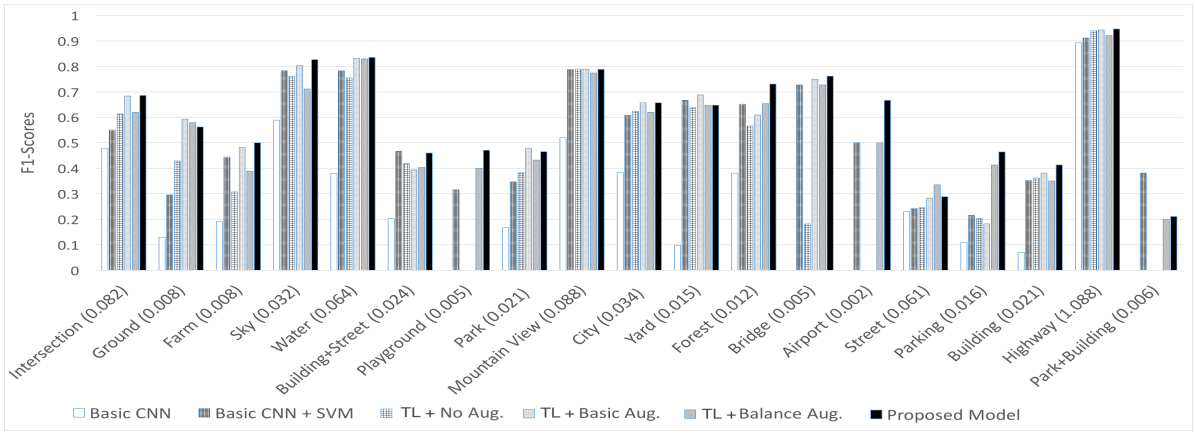

Figure 4. Comparison of F1-scores for each concept in the dataset.

the database. However, some cameras are offline and return black/unavailable images. Therefore, a check program traverses the directory to remove the bad quality images based on their byte sizes. The final cleaned dataset contains over 10,000 images captured from network cameras. Those images include 19 semantic concepts (scenes) such as highway, intersection, yard, and mountain. The dataset is carefully divided into 70% training, 10% reference, and 20% testing so that each set includes different samples from all classes. All the images are resized to 299*299 pixels. In this dataset, the P/N ratio for each concept ranges from 1.088 (concept "highway") to as low as 0.002 (concept "airport"), which leads to an imbalanced data classification problem.

4.2 Result Evaluation

The F1-score (Avg. F1) is adopted as the main evaluation metric since it is the most valuable comparison metric for imbalanced data and is the trade-offs between precision and recall. Moreover, the Weighted Average F1-score (WAvg. F1) and top-1 accuracy (Acc.) metrics are used to show that the proposed model can improve not only the prediction of individual minority classes but also the overall performance results. Here, WAvg. F1 is the average of the F1-scores of each class times its ratio of positive to all samples. The results of the proposed network are compared with the following models: (1) "Basic CNN": a model based on VGGNet [10] running from scratch on our dataset (deeper models such as Inception will not converge well on this dataset); (2) "Deep CNN features+SVM": using a deep CNN model as a fixed feature extractor and a linear support vector machine as a classifier; (3) "TL+No Aug.": a fine-tuned CNN model without data augmentation; (4) "TL+Basic Aug.": a fine-tuned model with real-time data augmentation; and finally (5) "TL+Balanced Aug.": a fine-tuned CNN model plus a modified data augmentation in which each training batch includes a balanced number of classes. This model utilizes both oversampling and undersampling techniques. In all transfer learning models, InceptionV3 is used as the base CNN model. Stochastic Gradient Descent (SGD) [9] is used as the optimization with learning rate 0.0001 and momentum 0.9. The "ImageDataGenerator" layer in Keras [3] is used for augmentation. Specifically, the augmentation parameters used in this paper are: shear_range=0.2, horizontal_flip=True, rotation_range=10, width_shift_range=0.2, and height_shift_range=0.2. Moreover, the threshold of model fusion, Tr, is selected as 0.3 based on the model performance on the reference data.

Table 1 illustrates the detailed performance results on this dataset. As can be inferred from the table, training a CNN from scratch performs the worst for all three evaluation metrics. This is due to the need for large-scale datasets to accurately update the random weights in CNNs. Transfer learning can significantly improve the results compared to basic CNN as shown in the third row of the table. However, it still cannot handle imbalanced data precisely. Similarly, the "TL+No Aug." model performs poorly on the dataset regarding the Avg. F1-score. However, the model "TL+Basic Aug." increases all the metrics compared to the no augmentation model. The "TL+Balanced Aug." model (a hybrid oversampling and undersampling model) can obviously improve the Avg. F1-score; however, its Acc. and WAvg. F1-scores are less than the ones in the original augmentation model. In other words, conventional imbalanced data techniques boost the performance of the minority classes by sacrificing the majority ones. Finally, the last row of the table shows how the proposed method improves the performance results for all three evaluation metrics. That is, it improves the prediction performance of the minority classes and also maintains the average accuracy.

The visualized performance results are demonstrated in Figure 4 which shows each concept (class) along with its

Table 1. Performance evaluation

Model	Acc.	Avg. F1	WAvg. F1
Basic CNN	0.649	0.254	0.630
Deep CNN Features+SVM	0.746	0.528	0.747
TL+No Aug.	0.765	0.432	0.755
TL+Basic Aug.	0.792	0.502	0.779
TL+Balanced Aug.	0.759	0.553	0.766
Proposed Model	0.802	0.599	0.794

P/N ratio in the parentheses. As can be seen from this figure, the distribution of the data is highly skewed. Several concepts have very low P/N ratios (e.g., airport, bridge, and playground), few concepts have higher P/N ratios (e.g., mountain view, intersection, and water), and the "highway" concept has a very high P/N ratio. "Basic CNN" has the lowest F1-score in all classes and cannot detect any instances in classes with very low P/N ratios. The "TL+No Aug." model improves the results compared to "Basic CNN", but it still cannot detect the minority classes. "Deep CNN features+SVM" performs better than all other models in concept "Park+Building", while it performs poorly in almost all other classes. "TL+Balance Aug." can detect some instances in the minority classes, though its performance is much lower than the "TL+Basic Aug." model in other classes (e.g., highway and intersection). Despite the good performance of the "TL+Basic Aug." model in some concepts (e.g., yard and street), it cannot detect any instances in the low P/N ratio classes. Finally, the proposed model can significantly improve the detection performance of the minority classes, while maintaining or even improving the performance of all other concepts. This shows the effectiveness of the proposed model to classify imbalanced and heterogeneous data from the real-world datasets.

5 Conclusion

This paper presents a multimedia system that is able to discover network cameras whose data is publicly available on the Internet, retrieve real-time visual data from these cameras, and analyze such imbalanced data for semantic concept detection. The semantic concept detection is achieved by the proposed dynamic sampling model which is based on CNNs together with real-time data augmentation to enhance the performance results for both minority and majority classes. The experimental results demonstrate the ability and effectiveness of the proposed model on real-time visual data captured by our multimedia system in detecting semantic concepts, especially for imbalanced data.

Acknowledgments

For Yung-Hsiang Lu, this project is supported in part by NSF ACI-1535108. For Shu-Ching Chen, it is supported in part by NSF CNS-1461926. The authors would like to thank the organizations that provide the camera data[2]. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] P. Cao, D. Zhao, and O. Zaiane. An optimized cost-sensitive svm for imbalanced data learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 280–292. Springer, 2013.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[3] F. Chollet. Keras (2015). `http://keras.io`, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] N. Jacobs, W. Burgin, R. Speyer, D. Ross, and R. Pless. Adventures in Archiving and Using Three Years of Webcam Images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 39–46, 2009.

[6] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[7] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.

[8] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar. Multimedia big data analytics: A survey. *ACM Computing Surveys*, 51(1), Article 10, 2017.

[9] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[11] A. Stepanchuk and H. Chu. RTMPDump. `https://rtmpdump.mplayerhq.hu/`, 2016.

[12] Y. Sun, A. K. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[14] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen. Deep learning for imbalanced multimedia data classification. In *IEEE International Symposium on Multimedia*, pages 483–488, 2015.

[2] A complete list of the data sources is available at https://www.cam2project.net/ack