A Distributed Automatic Video Annotation Platform

Md Anwarul Islam *, Md Azher Uddin and Young-Koo Lee *

Data and Knowledge Engineering Lab, Department of Computer Science and Engineering, Kyung Hee University, Suwon 446-701, Korea; azher@khu.ac.kr
* Correspondence: abir@khu.ac.kr (M.A.I.); yklee@khu.ac.kr (Y.-K.L.)

Received: 12 June 2020; Accepted: 29 July 2020; Published: 31 July 2020

Featured Application: This work can be applied for automatic video annotation.

Abstract: In the era of digital devices and the Internet, thousands of videos are taken and share through the Internet. Similarly, CCTV cameras in the digital city produce a large amount of video data that carry essential information. To handle the increased video data and generate knowledge, there is an increasing demand for distributed video annotation. Therefore, in this paper, we propose a novel distributed video annotation platform that explores the spatial information and temporal information. Afterward, we provide higher-level semantic information. The proposed framework is divided into two parts: spatial annotation and spatiotemporal annotation. Therefore, we propose a spatiotemporal descriptor, namely, volume local directional ternary pattern-three orthogonal planes (VLDTP–TOP) in a distributed manner using Spark. Moreover, we developed several state-of-the-art appearance-based and spatiotemporal-based feature descriptors on top of Spark. We also provide the distributed video annotation services for the end-users so that they can easily use the video annotation and APIs for development to produce new video annotation algorithms. Due to the lack of a spatiotemporal video annotation dataset that provides ground truth for both spatial and temporal information, we introduce a video annotation dataset, namely, STAD which provides ground truth for spatial and temporal information. An extensive experimental analysis was performed in order to validate the performance and scalability of the proposed feature descriptors, which proved the excellence of our proposed approach.

Keywords: video annotation; video processing; big data; distributed video annotation; Apache Spark; spatiotemporal video annotation; video retrieval

1. Introduction

With the rapid advances of the Internet and digital devices (e.g., mobiles and video cameras) millions of videos are being taken and shared on the Internet through social media. For instance, on YouTube, more than 250 h of videos are uploaded every minute, and over 25 million visitors see more than 4 billion videos per day. Due to the increase of video data on the Internet, automated technologies to retrieve the information from the videos are required. For example, textual information plays a significant role in content-based video retrieval [1,2], searching, browsing [3], and semantic indexing. Therefore, researchers worked on automated video annotation which aims to assign tags that belong to the frame of a video.

To provide video annotation, several works have been presented [4–8]. Video annotation reduces the semantic gap between low-level features and high-level semantics. In the literature, video annotation annotates context, which can be categorized into two categories: the visual context and the semantic context. The visual context in video annotation provides visual information that is relevant to the frame of a video. In [8–10] single or multiple tags/labels were annotated for a frame.
according to the relevant information. However, they did not consider semantic video information. In order to improve the video annotation techniques, [4,5] provided semantic information with the visual context which led to higher performance in terms of video retrieval, searching, and browsing. However, to obtain the semantic context, post-processing is required, such as the re-ranking method or pseudo-relevance feedback. In [5], the author used the conditional random field to gain the semantic video context. However, they ignored the weak semantic parts of information which caused loose essential semantic information. Moreover, this post-processing is complex and hard to calculate. Furthermore, the video annotation process is time consuming, especially the feature extraction part. In order to reduce the processing time, several studies have been done [11–15] wherein the author performed large-scale video processing in a distributed manner. However, these works did not consider distributed video annotation.

In order to overcome the aforementioned issues, we introduce and develop a spatiotemporal-based distributed video annotation platform. Our works provide both visual information (spatial information) and spatiotemporal (semantic) information. We divide our work into two parts: spatial annotation and spatiotemporal annotation. Each part further divided into three parts: preprocessing, feature extraction, and tag retrieval. For feature extraction, we introduced a spatiotemporal descriptor, namely, volume local directional ternary pattern-three orthogonal planes (VLDTP–TOP). Furthermore, we developed the distributed version of the existing state-of-the-art algorithms: LBP, LTP, and LDTP for the spatial feature extraction, and MBP, VLBP, and LBP–TOP for the temporal feature extraction. We employed open-source Apache Spark to distribute the feature extraction algorithms to make the platform scalable, faster, and fault-tolerant. In addition, we provide video annotation APIs for the developer. Since in this work, we propose a spatiotemporal-based video annotation platform, we required a new video dataset that contained spatial and temporal ground truth tags. Therefore, we introduce a new video dataset, the SpatioTemporal video Annotation Dataset (STAD). Experimental analysis was conducted on STAD, which showed the excellence of our proposed approach.

The main contributions of our works are:

- We introduce a new distributed spatiotemporal-based video annotation platform that provides visual information and spatiotemporal information of a video.
- Moreover, we propose a spatiotemporal feature descriptor named volume local directional ternary pattern-three orthogonal planes (VLDTP–TOP) on top of Spark.
- We developed several state-of-the-art appearance and spatiotemporal-based video annotation APIs for the developer and video annotation services for the end-users.
- Furthermore, we propose a video dataset that supports both spatial and spatiotemporal ground truth.
- Extensive experiments have been done to validate the performance, scalability, and effectiveness.

The rest of the paper is organized as follows: Existing works for video annotation are presented in Section 2. In Section 3, the details of our proposed video annotation platform and video annotation APIs are discussed. Section 4, briefly describes the proposed spatiotemporal video annotation platform, including the proposed spatiotemporal descriptor. Experimental analysis and comparisons with state-of-the-art are reported in Section 5. Finally, Section 6 summarizes the proposed approach.

2. Related Work

Currently, video annotation is one of the most fascinating topics among researchers, since it plays a vital role in the knowledge-based system (KBS) for generating knowledge by describing the content and essential video information; therefore, researchers have done a lot of work in this area. This section presents the literature review for video annotation.

Recent works [16–19] tried to exploit the contextual information from an image by detecting the object and scene information. YoloV3 [16] provided the fastest annotation techniques; they trained their network on deep convolutional neural network Darknet19 using image datasets COCO and PASCAL. Although all of these methods provided state-of-the-art object detection performance, they focused
on tagging the objects only. Annotation inherits the property of object detection and uses the concept of semantic image information by giving the textual context. Cheng et al. [20], divided automatic image annotation (AIA) into five categories according to the work done previously; model building, accuracy, and computational time were the ultimate targets for automatic video annotation (AVA). Among these techniques, the nearest neighbor approach is very popular. The author in [21,22] explored the automatic image annotation using the nearest neighbor technique by detecting the object that the image contained. However, video annotation is different, since the video contains spatiotemporal information and redundant spatial information.

Existing video annotation can be classified into: visual context annotation and semantic context annotation. In the spatial context annotation, researchers have exploited the natural property from each frame of a video. Visual context is assigned to each frame of a video in [7,8,23–29]. Here the authors in [26] exploited the multi-level visual context from each frame of a video using the nearest neighbor approach. Similarly, in [30,31] the authors explored the concept from a video and provided multi-level annotation. Later on, tree-based techniques were used in [32] to detect the duplicated video using scene annotation.

However, all of these techniques lack the semantic information. Since a video contains more than visual information, several works [4,5,33–36] have been done to analyze the semantic information from a video. These works explore the semantic information from each frame of a video and tag the predefined label that belongs to the frame.

However, all of these works miss the temporal information of a video. Time-based information provides essential information about an event, such as accidents and cycling. In order to capture the temporal information, [34,37] provided an automatic video annotation technique that keeps the temporal information, including the visual-spatial information. To find the temporal information over the sequential frame of a video the author in [34] used the association rules to correlate the spatial and temporal concepts from consecutive frames. On the other hand, in [37] the author proposed a spatiotemporal-based annotation to get the position of an object from several frames. However, all of those works did not consider the event information. Furthermore, they did not support distributed processing.

In contrast, the authors in [38] provided a web-based video annotation platform that can annotate plain video information, such as visual information. In [39] a distributed framework for activity recognition from mobile data was presented. However, they did not consider the image and video information.

In order to solve the aforementioned issues, we introduce a distributed spatiotemporal-based automatic video annotation platform that provides both the spatial information and the temporal information. Moreover, we provide distributed video annotation APIs for the developer and a video annotation service for the end-users.

3. Video Annotation Platform

Video annotation became an interesting research area among researchers due to the demand for knowledge creation, content-based video retrieval, and video indexing. Annotation data can be used to describe the events and generate knowledge.

The architecture for the proposed distributed video annotation is presented in Figure 1 and consists of mainly four layers, the big data storage layer (BDSL), the distributed video data processing layer (DVDPL), the distributed video data mining layer (DVDMML), and the video annotation services layer (VASL) respectively. Furthermore, the framework was designed for three basic users according to their roles: end-users, developers, and administrators. The end-users use this platform for their application-oriented solution. In contrast, APIs are developed for the developer so that a developer can produce customized applications. The proposed platform provides algorithm-as-a-service(AaaS) for the developer so that they can develop annotation services and algorithms. The administrator
is responsible for providing services to the end-users and video annotation APIs to the developer. Moreover, the administrator manages the platform and the services.

3.1. Big Data Storage Layer

The big data storage layer (BDSL) provides APIs to store data required for video annotation. The BDSL was created using Open-Source Hadoop HDFS which provides distributed data storage for different types of files. In our work we divided BDSL into five sub-parts: HDFS input/output (HDFS IO), distributed video data storage (DVDS), feature data storage (FDS), output data storage (ODS), and trained model storage (TMS).

Data are stored in a distributed manner and retrieved through the HDFS IO to different storage units for different layers. Raw video data are stored in distributed video data storage (DVDS) through the HDFS IO. DVDS supports diverse formats of video, such as mp4 and AVI. The frame extraction process is a relatively less time-consuming process; therefore, extracted frames are not stored. In contrast, the trained model, extracted feature data, and output data, such as tags (e.g., JSON files) are stored in the respective storage units through the HDFS IO. Similarly, data can be retrieved through the HDFS IO to the respective storage units.

3.2. Distributed Video Data Processing Layer (DVDPL)

The key part of the proposed architecture is the video data processing layer, wherein video data are processed in a distributed manner and intermediate feature data for the video data mining layer are returned. The main reason for distributed processing is to increase the performance and latency concerning the time that it takes for processing. Since the video data processing layer takes the more time than the other layer, we performed distributed processing in the DVDPL. Open-source distributed framework Apache Spark was utilized for distributed processing. Apache Spark provides a
distributed in-memory based framework for data labels. However, they do not have any native support for video data. Furthermore, Spark does not provide any higher-level APIs for video processing. Existing literature for video annotation does not provide distributed services and only considers spatial information and ignores the most significant temporal information from a video. However, temporal information is required to find the superior knowledge.

Therefore, to address the above-mentioned issues, we introduce a distributed video processing layer that integrates OpenCV with Apache Spark. Our work supports the video processing for both spatial and temporal information. Furthermore, we developed the state-of-the-art feature descriptors in a distributed manner.

We divide the proposed DVDPL into three sub-parts: preprocessing, a spatial feature extractor, and a dynamic feature extractor. For preprocessing we employ RGBtoGray conversion, frame extraction, and so on. Several existing algorithms such as LBP, LDP, LTP, and LDTP were developed on top of Spark for spatial feature extraction. Moreover, we present three-channel-based color local directional ternary pattern (CLDTP). To extract the dynamic features, we developed the existing state-of-the-art algorithms such as VLBP, LBP–TOP, and MBP. Furthermore, we proposed a novel dynamic feature descriptor named volume local directional ternary pattern-three orthogonal planes (VLDTP–TOP). All the proposed feature descriptors show excellence in terms of accuracy, scalability, and throughput. The details of the proposed feature descriptors are discussed in the following section.

3.3. Distributed Video Data Mining Layer (DVDML)

The distributed video data mining layer (DVDML) is responsible for assigning tags by measuring the similarity using the nearest neighbor algorithm, which takes the trained model and test features as input. The test feature comes from the previous layer and the trained model retrieves from HDFS through the HDFS IO unit. Finally, the output is stored in HDFS as a JSON, CSV, or text file.

3.4. Video Annotation Service Layer (VASL)

The video annotation service layer is used to interact with the system using a Web interface. End-users provide the query video to the system and get the desired output through the video annotation service layer (VASL). The developer can use APIs from different layers using the VASL layer.

3.5. Video Annotation APIs

We have developed video annotation APIs for the developer so that they can develop video annotation services. We have four main categories for video annotation APIs: preprocessing, spatial feature extraction, dynamic feature extraction, and similarity measure APIs. The details of our proposed video annotation APIs are described in Table A1.

4. Spatiotemporal Based Video Annotation

Figure 2 illustrates the end-to-end flow of our proposed spatiotemporal-based video annotation platform. The proposed platform is divided into two main parts: spatial video annotation (SVA) and spatiotemporal video annotation (STVA). SVA is responsible for retrieving spatial information from each frame of a video, which is similar to existing video annotation works. In the literature [7,16] video annotation detects objects from each frame of a video and annotate the object information. However, it ignores the true video information that comes with respect to time. Thus, our works proposed a spatiotemporal-based video annotation to solve this issue. Both parts are further divided into three sub-parts: a feature extractor, a similarity measure, and an annotator.

We have developed several state-of-the-art algorithms for both spatial and spatiotemporal feature descriptors. We implemented the existing LBP [40], LDTP [41], LTP [42], and CNN model (e.g., VGGNet [43]) on top of Spark in order to support distributed processing. Moreover, we introduced three-channel-based color LDTP. Similarly, we also developed the existing VLBP [44], LBP–TOP [44], and MBP [45] for capturing spatiotemporal features. Furthermore, we also proposed the local
directional ternary pattern–three orthogonal planes (LDTP–TOP) and the volume local directional ternary pattern–three orthogonal planes (VLDTP–TOP) for obtaining the spatiotemporal information. Our proposed descriptors show better precision than the existing algorithms. In the subsection, we describe our proposed descriptors.

**4.1. Color Local Directional Ternary Pattern (CLDTP)**

LBP [40] captures texture features from the gray image; however, it ignores the color as well gradient information which may decrease the performance. According to Thomas [46] better results can be achieved if we can keep the three prominent features, i.e., color, shape, and texture information. Therefore, in [41], the authors extended the LDP [47] to LDTP (local directional ternary pattern), which captures the shape information by applying the gradient operation; however, they did not consider the color information. The author in [48] used color information with LBP; the LBP operation was performed on each color channel. Afterward, the obtained features from each channel were fused to create a single feature vector that provided better accuracy. Hence, in this work, we introduce the color local directional ternary pattern (CLDTP), which includes texture, shape, and color information.

The block diagram of our proposed CLDTP is presented in Figure 3 where each frame is divided into five parts—center, bottom, upper, left, and right—to capture the local and global information from a frame. Afterward, we derive the color information from each frame by dividing the RGB image into three color channels—red, green, and blue. Then each channel is fed to the Equations (1) and (2) to calculate the local directional ternary pattern (LDTP), upper and lower respectively. We use Equation (3) to calculate the mean ($\mu$) to remove noise from given $3 \times 3$ square patches. Then Equations (4) and (5) are used to calculate the gradient magnitude for the center pixel $\alpha$ and neighboring pixel $\beta$ respectively which contains the shape information of an image. Here, $I_C$ and $I_P$ are the center and neighbor pixels respectively.

Finally, the ternary pattern calculation of LDTP upper and LDTP lower are performed using Equations (1) and (2) respectively. Here, $ER_C$ represents the edge response for center pixel and $ER_P$ presents the edge response for the $p^{th}$ neighbor pixel. In this approach, we use a modified Frei–Chen mask to reduce the computation complexity illustrated in Figure 4. Since square root operation is costly, we remove the square root from the Frei–Chen mask, which leads to faster calculation without compromising the accuracy. The details of edge response (ER) computation are depicted in Figure 5. Then the resultant edge response (ER), including the gradient information, is used to calculate the ternary pattern using Equations (1) and (2). Finally, we concatenate the results of the three channels (red, green, and blue) to obtain the color local directional ternary pattern (CLDTP).

\[
CLDTPU_{p,R} = 2^8 \delta(a, ER_C) + \sum_{p=0}^{7} 2^p \cdot \delta(\beta_p, ER_P) \tag{1}
\]

\[
CLDTPL_{p,R} = 2^8 \eta(a, ER_C) + \sum_{p=0}^{7} 2^p \cdot \eta(\beta_p, ER_P) \tag{2}
\]
where:

\[ \mu = \frac{1}{9}(I_C + \sum_{p=0}^{7} I_p) \]  

(3)

\[ \alpha = \mu - I_C \]  

(4)

\[ \beta_p = I_p - I_c \]  

(5)

and

\[ \delta(x, y) = \begin{cases} 
1 & \text{if } x \leq 0 \text{ and } y \leq 0 \\
0 & \text{else} 
\end{cases} \]  

(6)

\[ \eta(x, y) = \begin{cases} 
1 & \text{if } x \geq 0 \text{ and } y \geq 0 \\
0 & \text{else} 
\end{cases} \]  

(7)

**Figure 3.** Block diagram of proposed color local directional ternary pattern (CLDTP).

**Figure 4.** (a) Modified Frei–Chen masks in eight directions; (b) 2nd derivative of the Gaussian filter.
4.2. Local Directional Ternary Pattern–Three Orthogonal Plane (LDTP–TOP)

LBP–TOP [44] can capture the dynamic features from the video data; however, it suffers from noise and is sensitive to lighting illumination as it uses LBP for its computation. In order to overcome the issues of LBP–TOP, we proposed a new descriptor, namely, local directional ternary pattern–three orthogonal plane (LDTP–TOP), which includes shape, texture, and motion information. LDTP captures the spatial features that are used for texture analysis and we adopt LDTP for the temporal feature extraction by using three orthogonal planes, XY, XT, and YT. We take three consecutive frames \(I_{t-1}, I_t, I_{t+1}\) at the same time interval. In order to preserve temporal information, we used space-time information YT and XT. Figure 6 shows an example of XY, YT, and XT computation. The resultant output feature size \(6 \times 2^P\) is much smaller than the feature produced by the VLBP: feature size \(2^{3P+2}\). Therefore the calculation becomes comparatively much easier. In this approach, we get the three orthogonal planes information from three consecutive frames. Then, we use the modified Frei–Chen mask represented in Figure 4 for calculating edge response for each XY, XT, and YT plane.

After deriving the edge response (ER) for the neighbor \((ER_p)\) pixels and center \((ER_C)\) pixel, we use Equations (8) and (9) to calculate the ternary information, LDTP–TOP upper and LDTP–TOP lower respectively. Figure 7 illustrates the details of proposed LDTP–TOP algorithm.

\[
LDTP - TOP_{U,P,R} = 2^8 \delta(\alpha, ER_C) + \sum_{p=0}^{7} 2^p \delta(\beta_p, ER_p) 
\]

(8)

\[
LDTP - TOP_{L,P,R} = 2^8 \eta(\alpha, ER_C) + \sum_{p=0}^{7} 2^p \eta(\beta_p, ER_p) 
\]

(9)

Here, we also compute the gradient magnitude \(\beta\) by subtracting the neighbor pixel from its center pixel value using Equation (5). Furthermore, we calculate mean \((\mu)\) value of \(3 \times 3\) neighbor patch in order to remove noise through Equation (3); then computation of center gradient magnitude \(\alpha\) is done using Equation (4). Afterward, both the edge response and gradient of center-pixel \(\alpha\), and neighbor-pixel \(\beta\) are fed into the Equations (8) and (9) for LDTP–TOP upper and LDTP–TOP lower computation respectively. Finally three LDTP–TOP upper and three LDTP–TOP lower from the XY, YT, and XT are concatenated for final feature generation using Equation (10).
LDTP – TOP = LDTP − TOPU_{P,R}(XY) || LDTP − TOPU_{P,R}(YT) || LDTP − TOPL_{P,R}(XT) || LDTP − TOPL_{P,R}(XY) || LDTP − TOPL_{P,R}(YT) || LDTP − TOPL_{P,R}(XT)  

\[(10)\]

| \(I_{x-1}\) | \(I_{x}\) | \(I_{x+1}\) |
|---|---|---|
| 68 43 27 | 104 113 100 | 106 113 112 |
| 75 43 28 | 132 140 116 | 114 129 135 |
| 81 45 32 | 141 152 135 | 104 126 140 |

**Figure 6.** Illustration of XY, YT and XT from three consecutive frames.

**Figure 7.** Illustration of local directional ternary pattern—three orthogonal planes (LDTP–TOP)

### 4.3. Spatiotemporal Local Directional Ternary Pattern (VLDTP)

However, by applying the LDTP–TOP, we may still lose some information, since it takes 18 pixels into account during the computation from 27 pixels of the three consecutive frames, e.g., it ignores 8 pixels of information from the previous and next frame. Furthermore, in the LDTP–TOP computation, several pixels are used multiple times, which brings in a redundancy issue. In order to resolve this problem, we introduce the volume local directional ternary pattern (VLDTP). We use the same three consecutive frames used for LDTP–TOP and compute the edge response for the neighbor pixels of the former frame \(ER_{Fp}\), current frame \(ER_{Cp}\), and the next frame \(ER_{Np}\) using the modified Frei–Chen mask and center edge response for the center pixel of the former frame \(ER_{FC}\), current frame \(ER_{CC}\), and the next frame \(ER_{NC}\) using the 2nd derivative Gaussian filter presented in Figure 4b. Afterward, we calculate VLDTP upper (VLDTPU) and VLDTP lower (VLDTPL) patterns for the former frame \(VLDTPU_{Fp}\), current frame \(VLDTPU_{Cp}\), and next frame \(VLDTPU_{Np}\) using Equations (11)–(16) respectively. The illustration of the proposed VLDTP is depicted in Figure 8.

\[
VLDTPU_{Fp,R} = 2^8 \delta(\alpha, ER_{FC}) + \sum_{p=0}^{7} 2^p \delta(\beta_p, ER_{FP})
\]

\[(11)\]
Finally, the max-pooling operation is employed in order to compute the final VLDTP upper (VLDTPU) and VLDTP lower (VLDTPL) patterns and we concatenate these two features to produce the final feature vector.

\[
\text{VLDTPU} = \text{Max}\left\{\text{VLDTPU}_{FP,R}, \text{VLDTPU}_{CP,R}, \text{VLDTPU}_{NP,R}\right\} \\
\text{VLDTPL} = \text{Max}\left\{\text{VLDTPL}_{FP,R}, \text{VLDTPL}_{CP,R}, \text{VLDTPL}_{NP,R}\right\}
\]

\[
\text{VLDTP} = \text{VLDTPU} \mid\mid \text{VLDTPL}
\]

Finally, the volume local directional ternary pattern (VLDTP) is illustrated in Figure 8.

4.4. Volume Local Directional Ternary Pattern–Three Orthogonal Plane (VLDTP–TOP)

Afterward, we compute the volume local directional ternary pattern–three orthogonal planes (VLDTP–TOP) by using a concatenated fusion of LDTP–TOP and VLDTP using Equation (20). Since we solve the LBP–TOP issues using LDTP–TOP and LDTP–TOP issues using VLDTP, the proposed VLDTP–TOP leads to better accuracy. On the other hand, feature extraction is the most time-consuming and complex part of our proposed framework; therefore, to enhance and boost the performance of our proposed approach we use Apache Spark to distribute feature extraction part that leads to faster processing and better throughput. Furthermore, we used distributed deep learning framework DeepLearning4J (DL4J) [49] to extract deep features, which uses the ND4J [50] framework. DL4j framework has a built-in deep learning model which was implemented in the distributed scenario.
\[ VLDTP - TOP = LDTP - TOP || VLDTP \] (20)

4.5. Similarity Measure

Finally, the obtained features from the feature descriptors are fed into the nearest neighbor algorithm. Here, the similarity measure is found between the trained model and the query data using the nearest neighbor algorithm and it returns the list of similar tags for a spatial feature and a single tag for the spatiotemporal feature. Since the nearest neighbor approach is costly, we perform the similarity measure in a distributed manner. As a result, it becomes more powerful and faster.

5. Evaluation and Analysis

In this section, the performance of the proposed work has been evaluated on the proposed STAD [51] and UCF50 [52] datasets. Here, we compute the scalability and throughput of the proposed APIs. Furthermore, we also calculated the precision, and average precision (AP) for automatic video annotation. Precision describes how accurately the prediction provided by an algorithm. The precision calculation has been done using Equation (21). Precision (P) is calculated by the true positive (TP) over the summation of true positive (TP) and false positive (FP). We calculated precision for each category. Afterward, we computed average precision (AP) by taking the ratio between precision and number of categories using Equation (22). To compare the performance of different the feature descriptors, we used average precision (AP).

\[
Prec = \frac{TP}{TP + FP} \\
AP = \frac{\sum_{n=1}^{N} P(n)}{N}
\]

5.1. Experimental Setup

For video annotation, we built a distributed platform using the Hortonworks Data Platform (HDP 3.1.0) consisting of four machines. HDP integrates several distributed open-source tools—Hadoop, HDFS, Hbase, Apache Spark, and so forth. Each node is connected through the fully-managed switch GSM7328S which delivers 24 and 48 ports for high-density copper connectivity of auto-sensing 1000 Mbps. The cluster’s structure and specifications are illustrated in Figure 9. Each node in Figure 9 has four parameters, such as Cent OS 7.6 | i5 | 128GB | 13 TB. These parameters indicate the operating system, processor model, memory size in GB, and data storage size in TB respectively.

Figure 9. Distributed platform for experimental analysis.
5.2. STAD Dataset

Existing video datasets consider either spatial or temporal information. However, there are no datasets available that provide both. Due to the lack of a spatiotemporal video dataset for video annotation domain that contains ground truth information both for spatial and temporal information, we proposed a novel spatiotemporal video dataset STAD [51] which includes diverse categories, as depicted in Figure 10. STAD [51] consists of 11 dynamic and 20 appearance categories. Among dynamic categories, there are four main categories: human action, emergency, traffic, and nature. We divide human action into two subcategories: single movement and crowd movement. The STAD dataset is the combination of UCF101 [53], Dynamic texture DB [54], and a subset of YouTube 8M [55]. The details of our proposed STAD dataset are given in Table 1.

![Sample frames of the proposed STAD video dataset.](image)

Table 1. Description of proposed STAD video dataset.

| Category         | Sub-Category          | Dynamic Information | Spatial Appearances          |
|------------------|-----------------------|---------------------|------------------------------|
| Human Action     | Single Movement       |                     |                              |
|                  | BaseballPitch         | Person, Ball, Field | Grass/Green_field, Sky      |
|                  | Biking                | Person, Cycle, Tree | Car, Road, Fence, Sky, Grass/Green_field |
|                  | Horse Riding          | Person, Horse, Tree | Bush, Fence, Sky, Grass/Green_field |
|                  | Skate Boarding        | Person, Road, Tree  | Sky                          |
|                  | Swing                 | Person, Tree, Bush  | Grass/Green_field            |
| Crowd Movement   | Band Marching         | Person, Road, Tree  | Sky, Building                |
|                  | Run                   | Person, Road, Tree  | Sky, Building, Car           |
| Emergency        | Explosions            | Person, Fire, Building, Sky, Hill, Fire, Volcano |
|                  | Tornado               | Tornado, Grass/Green_field, Stamp |
| Traffic          | Car                   | Car, Road, Sky, Tree | Bush, Building, Grass/Green_field |
| Nature           | Birds Fly             | Birds, Sky, Person, Building, Sea, Grass/Green_field |

5.3. Experimental Analysis

Figure 11 shows the scalability of the proposed distributed video annotation APIs. We have developed and implemented distributed feature extraction APIs. For instance, local binary pattern (LBP) [40], local directional ternary pattern (LDTP) [41], local ternary pattern (LTP) [42], color local binary pattern (CLBP), and color local directional ternary pattern (CLDTP) were developed for spatial feature extraction, and volume local binary pattern (VLBP) [44], local binary pattern–three
orthogonal planes (LBP–TOP) [44], motion binary pattern (MBP) [45], and volume local directional ternary pattern–three orthogonal planes (VLDTP–TOP) were developed for dynamic feature extraction respectively. Furthermore, we have used the distributed deep learning for Java (DL4J) [49] for developing the CNN APIs. We also implemented a distributed similarity measure nearest neighbor algorithm for both the spatial and temporal features.

Scalability has been tested on the proposed STAD dataset. Figure 11 shows that for one node it takes more time than employing three nodes. With the increase of nodes, the processing time decreases, which proves the scalability of the platform. From Figure 11a we can observe that, for the temporal feature extraction, MBP takes less time than the other descriptors in all three cases. MBP takes 213 s in one node and 64 s in three nodes. The proposed VLDTP–TOP takes more time than the existing descriptors due to the computation complexity. VLDTP–TOP takes 778 s in one node and its performance is increased when three nodes are employed. However, VLDTP–TOP outperforms in terms of average precision compared the existing descriptors. The LBP-based texture feature descriptor takes less time than other descriptors due to the simplicity of the algorithm. LBP took 97 s in one node while its performance was improved when three nodes were utilized and it took only 30 s. CLDTP with the original mask took 251 s with one node and 84 s for three nodes. However, the proposed CLDTP with a modified Frei–Chen mask took 183 s in one node and performance increased when using three nodes. Figure 11c shows that the performance of the nearest neighbor algorithm increased with the number of nodes. For one node the the nearest neighbor algorithm took 243 s (temporal) and 66.8 s (spatial), whereas for three nodes it took 95 s and 25 s for the temporal and spatial annotation respectively.

![Figure 11. Scalability of the low-level APIs for distributed video annotation on the STAD dataset. (a) Time (in seconds) required for the dynamic feature extraction and (b) time (in seconds) required for the spatial feature extraction. (c) time (in seconds) required for the similarity measure APIs.](image-url)
Furthermore, we also computed the throughput of our proposed APIs, which is demonstrated in Figure 12a,b. This experiment shows that the throughput of our proposed APIs is increased if the number of nodes increases. From Figure 12a we can observe that, for temporal feature extraction, the proposed VLDTP–TOP can process 154 frames per second in one node and process 408 frames per second when three nodes are used. MBP can process the maximum number of frames among all the dynamic descriptors. Moreover, Figure 12 shows that our descriptor can achieve more than real-time performance with the current platform. Figure 12b shows that the proposed CLDTP can process 573 frames in one second and perform faster than real-time capture when three nodes are applied. However, LBP can process the maximum number of frames among all the spatial descriptors due to computational simplicity.

Figure 12. (a) Throughput (in seconds) of the temporal feature extraction APIs on STAD dataset and (b) throughput (in seconds) of the spatial feature extraction APIs on the STAD dataset.

Figure 13 demonstrates the precision for each category of the proposed color local directional ternary pattern (CLDTP) and VGGNet on STAD datasets. Though the AP of VGGNet shows better than our proposed CLDTP, CLDTP still works better for roads, fences, and tornadoes. Furthermore, Figure 14 illustrates that CLDTP is faster than vggNet for feature extraction.

Figure 13. Precision of the CLDTP and VGGNet on STAD.
Figure 14. Time required for feature extraction using CLDTP and VGGNet on STAD.

Figure 15 represents the average precision (AP) of our proposed spatial feature extraction APIs. The comparison shows that our proposed descriptor provides better performance than the existing handcrafted algorithms: LBP, LTP, CLBP, LBPC [48], and DFT [8]. The existing descriptors only capture either the textural features or color features and ignore the shape information. In this work, we extracted color information, including the shape and texture information; therefore, our proposed CLDTP provides better performance than the existing descriptors. Thomas et al. [46] presented that accuracy will be improved if three prominent features, e.g., color, texture, and shape features, are preserved. However, CLDTP shows lower precision than the deep network VGGNet in some categories.

Figure 15. Average precision (AP) of spatial feature extraction APIs on the STAD dataset.

Figure 16 illustrates the precision of each category for the proposed temporal feature descriptor VLDTP–TOP and the existing dynamic descriptors MBP and LBP–TOP on the STAD dataset. Proposed VLDTP–TOP performs better in almost every category. However, VLDTP–TOP cannot differentiate between smoke in explosions and tornados properly.

Figure 17 demonstrates the average precision for the proposed temporal feature extraction APIs. LBP–TOP suffers from noise and is sensitive to illumination as it uses LBP for its computation and our proposed LDTP–TOP removes noises using mean deviation and 2nd derivative Gaussian filter in the center pixel. Furthermore, we capture the gradient information with the dynamic texture, which leads to higher accuracy than LBP–TOP. Moreover, both LBP–TOP and LDTP–TOP use only 18 pixels out of
27 pixels and use redundant information for some pixels. Therefore, it loses some crucial information, which leads to lower performance. To solve this problem, VLDTP–TOP considers volume information; it takes similar frame information for its computation to that used to discover orthogonal planes. We keep only the maximal information from both upper and lower patterns since max-pooling does not loose information, but rather, it preserves important features. Therefore, VLDTP–TOP outperforms the existing dynamic descriptors.

![Figure 16](image1.png)

**Figure 16.** Precision of each category for MBP, LBP–TOP, and VLDTP–TOP descriptors on the STAD dataset.

![Figure 17](image2.png)

**Figure 17.** Average precision (AP) of temporal feature extraction APIs on the STAD dataset.

In order to validate the performance of the proposed VLDTP–TOP on larger dynamic dataset, an experiment on UCF50 [52] was performed. Figure 18 shows that the proposed VLDTP–TOP presents better performance than most of the existing approaches. However, iDT [56] obtained the best performance among them all.
Figure 18. Comparison between the proposed approach and the extant approaches for the UCF50 dataset.

6. Conclusions

In this paper, we proposed a novel distributed spatiotemporal-based video annotation platform that captures spatial and temporal information by utilizing in-memory based computing technology Apache Spark that works on top of Hadoop HDFS. Furthermore, we proposed a distributed dynamic feature descriptor, volume local directional ternary pattern–three orthogonal planes (VLDTP–TOP), which has better performance than the existing algorithms. We also implemented several state-of-the-art algorithms. Lastly, we proposed a diverse and complex dataset, STAD. Our provided services are robust, scalable, and efficient.

Author Contributions: Y.-K.L. guided improvement through the discussion; M.A.I. conceived the key idea, performed implementation, and was in charge of writing the manuscript. M.A.U. contributed to the idea optimization, updating of the manuscript, and revision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (number 2016-0-00406, SIATCCTV Cloud Platform).

Acknowledgments: This work was supported by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (number 2016-0-00406, SIATCCTV Cloud Platform).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proposed Video Annotation APIs

| Category              | APIs                                                                 | Description                                                                 |
|-----------------------|----------------------------------------------------------------------|------------------------------------------------------------------------------|
| Pre-processing        | siat.vdpl.videoAnnotation.preprocessing.segment(frame, size)         | Segment a given frame from a frame according to given number of segment       |
|                       | siat.vdpl.videoAnnotation.preprocessing.frameExtract(video)          | Extract each frame from a video                                              |
|                       | siat.vdpl.videoAnnotation.preprocessing.rgb2gray(frame)              | Convert RGB frame to gray frame                                              |
| Spatial Feature Extraction | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.AlexNet(Mat frame) | Capture the deep spatial feature from each frame of a video based on AlexNet |
|                       | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.VGGNet(Mat frame) | Capture the deep spatial feature from each frame of a video based on VGG16    |
|                       | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.lbp(frame, neighbor, radius) | Capture the spatial texture feature from each frame of a video based on LBP |
Table A1. Cont.

| Category | APIs                                                                 | Description                                                                                     |
|----------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|          | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.ltp(frame,      | Capture the spatial texture feature from each frame of a video based on Local Ternary Pattern(LTP) |
|          | neighbor, radius)                                                    |                                                                                                |
|          | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.clbp(frame,     | Capture the color spatial feature from each frame of a video based on Color Local Binary Pattern(CLBP) |
|          | neighbor, radius)                                                    |                                                                                                |
|          | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.ldtp(frame,     | Capture the color spatial feature from each frame of a video based on Local Directional Ternary Pattern(LDTP) |
|          | neighbor, radius)                                                    |                                                                                                |
|          | siat.vdpl.videoAnnotation.sceneBasedFeatureExtractor.cldtp(frame,    | Capture the color spatial feature from each frame of a video based on Color Local Directional Ternary Pattern(CLDTP) |
|          | neighbor, radius)                                                    |                                                                                                |
| Dynamic Feature Extraction | siat.vdpl.videoAnnotation.dynamicFeatureExtractor.VLBC (FormerFrame, CurrentFrame,NextFrame, neighbor, radius) | Capture the dynamic texture feature from each video based on Volume Local Binary Count(VLBC) |
| | siat.vdpl.videoAnnotation.dynamicFeatureExtractor.VLBP (FormerFrame, CurrentFrame,NextFrame, neighbor, radius) | Capture the dynamic texture feature from each video based on Volume Local Binary Pattern (VLBP) |
| | siat.vdpl.videoAnnotation.dynamicFeatureExtractor.VLTrP (FormerFrame, CurrentFrame,NextFrame, neighbor, radius) | Capture the dynamic texture feature from each video based on Volume Local Transition Pattern (VLTrP) |
|          | siat.vdpl.videoAnnotation.dynamicFeatureExtractor.VLDTOP-TOP (FormerFrame, CurrentFrame,NextFrame, neighbor, radius) | Capture the dynamic texture feature from each video based on Volume Local Directional Ternary Pattern-Three Orthogonal Planes (VDTP-TOP) |
| Similarity Measure | siat.vdml.videoAnnotation.SearchSimilarFrame.nearestNeighbor (feature) | Measure similar frames between train model and test frames |
|          | siat.vdml.videoAnnotation.SearchSimilarClip.nearestNeighbor (feature) | Measure similar video between train model and test video |

References

1. Peng, Y.; Ngo, C.W. Clip-based similarity measure for query-dependent clip retrieval and video summarization. IEEE Trans. Circuits Syst. Video Technol. 2006, 16, 612–627. [CrossRef]
2. Lew, M.S.; Sebe, N.; Djeraba, C.; Jain, R. Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimed. Comput. Commun. Appl. 2006, 2, 1–19. [CrossRef]
3. Smith, J.R. VideoZoom spatio-temporal video browser. IEEE Trans. Multimed. 1999, 1, 157–171. [CrossRef]
4. Jiang, Y.G.; Dai, Q.; Wang, J.; Ngo, C.W.; Xue, X.; Chang, S.F. Fast semantic diffusion for large-scale context-based image and video annotation. IEEE Trans. Image Process. 2012, 21, 3080–3091. [CrossRef] [PubMed]
5. Yi, J.; Peng, Y.; Xiao, J. Exploiting semantic and visual context for effective video annotation. IEEE Trans. Multimed. 2015, 15, 1400–1414. [CrossRef]
6. Vondrick, C.; Patterson, D.; Ramanan, D. Efficiently scaling up crowdsourced video annotation. Int. J. Comput. Vis. 2013, 101, 184–204. [CrossRef]
7. Markatopoulou, F.; Mezaris, V.; Patras, I. Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. IEEE Trans. Circuits Syst. Video Technol. 2018, 29, 1631–1644. [CrossRef]
8. Aote, S.S.; Potnurwar, A. An automatic video annotation framework based on two level keyframe extraction mechanism. Multimed. Tools Appl. 2019, 78, 14465–14484. [CrossRef]
9. Snoek, C.G.; Worring, M.; Van Gemert, J.C.; Geusebroek, J.M.; Smeulders, A.W. The challenge problem for automated detection of 101 semantic concepts in multimedia. In Proceedings of the 14th ACM international conference on Multimedia, Santa Barbara, CA, USA, 23–27 October 2006; pp. 421–430.
10. Jiang, Y.G.; Yang, J.; Ngo, C.W.; Hauptmann, A.G. Representations of keypoint-based semantic concept detection: A comprehensive study. IEEE Trans. Multimed. 2009, 12, 42–53. [CrossRef]
11. Nazare, A.C., Jr; Schwartz, W.R. A scalable and flexible framework for smart video surveillance. Comput. Vis. Image Underst. 2016, 144, 258–275. [CrossRef]
12. Tu, N.A.; Huynh-The, T.; Lee, Y. Scalable Video Classification using Bag of Visual Words on Spark. In Proceedings of the 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2–4 December 2019; pp. 1–8.

13. Uddin, M.A.; Alam, A.; Tu, N.A.; Islam, M.S.; Lee, Y.K. SIAT: A distributed video analytics framework for intelligent video surveillance. *Symmetry* 2019, 11, 911. [CrossRef]

14. Shidik, G.F.; Noersasongko, E.; Nugraha, A.; Andono, P.N.; Jumanto, J.; Kusuma, E.J. A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets. *IEEE Access* 2019, 7, 170457–170473. [CrossRef]

15. Li, D.; Zhang, Z.; Yu, K.; Huang, K.; Tan, T. ISEE: An Intelligent Scene Exploration and Evaluation Platform for Large-Scale Visual Surveillance. *IEEE Trans. Parallel Distrib. Syst.* 2019, 30, 2743–2758. [CrossRef]

16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.

17. Jiang, S.; Liang, S.; Chen, C.; Zhu, Y.; Li, X. Class agnostic image common object detection. *IEEE Trans. Image Process.* 2019, 28, 2836–2846. [CrossRef] [PubMed]

18. Luo, A.; Li, X.; Yang, F.; Jiao, Z.; Cheng, H. Webly-supervised learning for salient object detection. *Pattern Recognit.* 2020, 103, 107308. [CrossRef]

19. Zhang, Q.; Shi, Y.; Zhang, X. Attention and boundary guided salient object detection. *Pattern Recognit.* 2020, 107, 107484. [CrossRef]

20. Cheng, Q.; Zhang, Q.; Fu, P.; Tu, C.; Li, S. A survey and analysis on automatic image annotation. *Pattern Recognit.* 2018, 79, 242–259. [CrossRef]

21. Makadia, A.; Pavlovic, V.; Kumar, S. A new baseline for image annotation. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 316–329.

22. Guillaumin, M.; Mensink, T.; Verbeek, J.; Schmid, C. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 309–316.

23. Dorado, A.; Calic, J.; Izquierdo, E. A rule-based video annotation system. *IEEE Trans. Circuits Syst. Video Technol.* 2004, 14, 622–633. [CrossRef]

24. Qi, G.J.; Hua, X.S.; Rui, Y.; Tang, J.; Mei, T.; Zhang, H.J. Correlative multi-label video annotation. In Proceedings of the 15th ACM International Conference on Multimedia, New York, NY, USA, 10–16 October 2007; pp. 17–26.

25. Tang, J.; Hua, X.S.; Wang, M.; Gu, Z.; Qi, G.J.; Wu, X. Correlative linear neighborhood propagation for video annotation. *IEEE Trans. Syst. Man, Cybern. Part B* 2008, 39, 409–416. [CrossRef]

26. Wang, M.; Hua, X.S.; Tang, J.; Hong, R. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *IEEE Trans. Multimedia.* 2009, 11, 465–476. [CrossRef]

27. Wang, M.; Hua, X.S.; Hong, R.; Tang, J.; Qi, G.J.; Song, Y. Unified video annotation via multigraph learning. *IEEE Trans. Circuits Syst. Video Technol.* 2009, 19, 733–746. [CrossRef]

28. Lai, J.L.; Yi, Y. Key frame extraction based on visual attention model. *J. Vis. Commun. Image Represent.* 2012, 23, 114–125. [CrossRef]

29. Thakar, V.B.; Hadia, S.K. An adaptive novel feature based approach for automatic video shot boundary detection. In Proceedings of the 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), Gujarat, India, 1–2 March 2013; pp. 145–149.

30. Xu, S.; Tang, S.; Zhang, Y.; Li, J.; Zheng, Y.T. Exploring multi-modality structure for cross domain adaptation in video concept annotation. *Neurocomputing* 2012, 95, 11–21. [CrossRef]

31. Chamasemani, F.F.; Affendey, L.S.; Mustapha, N.; Khalid, F. Automatic video annotation framework using concept detectors. *J. Appl. Sci.* 2015, 15, 256. [CrossRef]

32. Chou, C.L.; Chen, H.T.; Lee, S.Y. Multimodal video-to-near-scene annotation. *IEEE Trans. Multimedia.* 2016, 19, 354–366. [CrossRef]

33. Xu, C.; Wang, J.; Lu, H.; Zhang, Y. A novel framework for semantic annotation and personalized retrieval of sports video. *IEEE Trans. Multimedia.* 2008, 10, 421–436.

34. Li, Y.; Tian, Y.; Duan, L.Y.; Yang, J.; Huang, T.; Gao, W. Sequence multi-labeling: A unified video annotation scheme with spatial and temporal context. *IEEE Trans. Multimedia.* 2010, 12, 814–828. [CrossRef]

35. Altadmri, A.; Ahmed, A. A framework for automatic semantic video annotation. *Multimed. Tools Appl.* 2014, 72, 1167–1191. [CrossRef]
36. Wang, X.; Yan, G.; Wang, H.; Fu, J.; Hu, J.; Wang, J.; Yang, Y.; Zhang, G.; Bao, H. Semantic annotation for complex video street views based on 2D-3D multi-feature fusion and aggregated boosting decision forests. Pattern Recognit. 2017, 62, 189–201. [CrossRef]
37. Wang, W.C.; Chiou, C.Y.; Huang, C.R.; Chung, P.C.; Huang, W.Y. Spatiotemporal Coherence-Based Annotation Placement for Surveillance Videos. IEEE Trans. Circuits Syst. Video Technol. 2016, 28, 787–801. [CrossRef]
38. Kavasidis, I.; Palazzo, S.; Di Salvo, R.; Giordano, D.; Spampinato, C. An innovative web-based collaborative platform for video annotation. Multimed. Tools Appl. 2014, 70, 413–432. [CrossRef]
39. Wu, J.; Feng, Y.; Sun, P. Sensor fusion for recognition of activities of daily living. Sensors 2018, 18, 4029. [CrossRef] [PubMed]
40. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. Pattern Recognit. 1996, 29, 51–59. [CrossRef]
41. Chahi, A.; Ruichek, Y.; Touahni, R. Local directional ternary pattern: A new texture descriptor for texture classification. Comput. Vis. Image Underst. 2018, 169, 14–27.
42. Tan, X.; Triggs, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Process. 2010, 19, 1635–1650.
43. Karen, S.; Andrew, Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
44. Zhao, G.; Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. 2007, 29, 915–928. [CrossRef]
45. Baumann, F.; Lao, J.; Ehlers, A.; Rosenhahn, B. Motion Binary Patterns for Action Recognition. In Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM-2014), Angers, France, 6–8 March 2014; pp. 385–392.
46. Thomas, A.; Sreekumar, K. A survey on image feature descriptors-color, shape and texture. Int. J. Comput. Sci. Inf. Technol. 2014, 5, 7847–7850.
47. Jabid, T.; Kabir, M.H.; Chao, O. Local directional pattern (LDP) for face recognition. In Proceedings of the 2010 digest of technical papers international conference on consumer electronics (ICCE), Las Vegas, NV, USA, 9–13 January 2010; pp. 329–330.
48. Singh, C.; Walia, E.; Kaur, K.P. Color texture description with novel local binary patterns for effective image retrieval. Pattern Recognit. 2018, 76, 50–68. [CrossRef]
49. Team, E.D.D. Deeplearning4j: Open-Source Distributed Deep Learning for the JVM, Apache Software Foundation License 2.0. 2016. Available online: http://deeplearning4j.org (accessed on 20 January 2020).
50. Team, N.D. ND4J: N-Dimensional Arrays and Scientific Computing for the JVM, Apache Software Foundation License 2.0. 2016. Available online: http://nd4j.org (accessed on 20 January 2020).
51. Abir, M.A.I. STAD. 2020. Available online: https://drive.google.com/drive/folders/1AWUkdiLlNaKuYLtsxLpx3KRaAmdnKlyz59?usp=sharing (accessed on 20 May 2020).
52. Reddy, K.K.; Shah, M. Recognizing 50 human action categories of web videos. Mach. Vis. Appl. 2013, 24, 971–981. [CrossRef]
53. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv 2012, arXiv:1212.0402.
54. Hadji, I.; Wildes, R.P. A New Large Scale Dynamic Texture Dataset with Application to ConvNet Understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 320–335.
55. Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; Vijayanarasimhan, S. Youtube-8m: A large-scale video classification benchmark. arXiv 2016, arXiv:1609.08675.
56. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.