An E-sports Video Highlight Generator Using Win-Loss Probability Model

Seok-Kyu Kang
Sungkyunkwan University
South Korea
kittyhyen@skku.edu

Jee-Hyong Lee*
Sungkyunkwan University
South Korea
john@skku.edu

ABSTRACT

Highlighting, to compile key scenes in a match, is an essential task in the E-sports industry. Because E-sports is less dependent on time and place in its nature, there are countless matches every day. However, creating a highlight is intensive and time-consuming labor. Even the quality of the outcome depends on the editor's ability and decision. We propose a new approach to an automatic highlight generator from E-sports match videos. Our highlight generator reduces the cost of production and allows stable quality control. We defined the smallest component of the highlight as a 'Point', the moment when the win-loss probability of each team changes drastically. A 'Clip' is the contextual scene around Points, and a set of Clips is a 'Highlight'. Our highlight generator can prioritize Points, using a model that detects changes in real-time win-loss probabilities. It can create various versions of a highlight by adjusting the number of Points. Also, because it operates in real-time, we can generate highlights instantly. Our generator recognizes and extracts real-time state information from a match video using OpenCV and CNN. It uses an MLP model and gets the win-loss probability of each moment to calculate the change rate between directly adjacent moments. This MLP is pre-trained with match results in the past. A threshold is set by partially implementing CART algorithm for Gini Impurity. If a moment's change rate satisfies the threshold, it is classified as a Point. Then, a Clip is specified by setting the interval before and after the Point using the average interval. Finally, merging the Clips becomes a highlight. We created various versions of highlight for 119 match videos of E-Sports, the league of legend 2018 World Championship. To calculate the win-loss probability, the MLP pretrained the 11,082 match results. As a result, the accuracy and f1 scores were 89.9% and 74.5%, respectively, for the version most similar to the official highlight. We also compare our highlights by human evaluation. About 65% of reviewers said our highlights are better than official ones. In the exclusive Clip evaluations, the Clips exclusively included in ours earn 4.02 Point, but the Clips only in the official ones earn 2.63, which says our highlights include much better Clips than the official highlights.

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SAC '20, March 30-April 3, 2020, Brno, Czech Republic
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6866-7/20/03…$15.00
https://doi.org/10.1145/3341105.3373894

CCS CONCEPTS

• **Computing methodologies → Video summarization**; **Interest Point and salient region detections**; **Neural networks**; **Supervised learning by classification**.

KEYWORDS

E-sports Video Highlighting, Win-Loss probability, Video Summarization, Image Recognition, OCR

ACM Reference Format:
Seok-Kyu Kang and Jee-Hyong Lee. 2020. An E-sports Video Highlight Generator Using Win-Loss Probability Model. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341105.3373894

1 INTRODUCTION

E-sports, a video game competition, is popular sports these days. In the 2018 Asian Games, E-sports was adopted as an official sport. The number of viewers of 'League of Legends 2018 World Championship' reached 100 million. Highlighting, to compile key scenes in a match, is an essential task in the E-sports industry. Because E-sports is less dependent on time and place than original sports in its nature, countless matches are held every day. However, creating a highlight is intensive and time-consuming labor. Even the quality of the outcome depends on the editor's ability and decision. In local, amateur, or individual leagues of famous players, there are many demands for highlight, but in most cases, it is difficult to produce them due to cost and time issues.

There have been some researches for the automatic generation of E-sports [17, 18]. They tried to process the whole image in every frame of the entire match video and to predict whether each frame needed to be included in highlight or not. They also detected event messages or added natural language processing to audience chat reviews. These previous approaches take complex and heavy models. They also tried to predict frame by frame, so the highlight video they generated is discrete and unnatural as if over-fitting. Of course, they are slow, and their quality is not stable. In the paper, we propose a new approach to an automatic highlight generator from E-sports match videos. Our highlight generator is light and fast, so it produces highlight in real-time, and the output can be comparable to the official highlight produced by human experts. We focused on recognizing real-time state information for the dashboard and side images on the match video, not from the whole screen image, and detecting important moments by win-loss probability prediction models, not to evaluate all the frames. Our approach is based on the win-loss probability of each team in the middle of a match. If the probability of win-loss can be predicted in real-time, then the moments that these probabilities drastically change will be highlight Points. For example, the win-loss probability of each team is 50:50 at the start of a match. If there is an important moment that needs to be included in a highlight, the probability of win-loss of each team will change, like to 30:70. Also, we can prioritize those moments and choose more important ones. For example, there are two moments in a match. One moment changes the win-loss probability from 50:50 to 30:70, and the next moment changes it to 90:10. Then, we can infer that the second moment is more important than the first one. We collect the state information at the end of the match from the records of the 11,082 official matches held in the past and build a Multi-Layer Perceptron(MLP) model with high accuracy of classification prediction. We define this model as a 'Win-loss probability model'. We apply this pretrained model to the target matches and get the real-time win-loss probability for each team using the sigmoid of the output layer.

Our generator calculates the rate of change in win-loss probabilities in each of the moments of the target matches, then finds an optimal threshold for the rate of change. It designates the moment beyond that threshold as a 'Point' for the highlight. The model for finding this optimal threshold is called a 'Point model' separately. It learns through a training set, which partially implements the Gini Impurity of the Classification and Regression Tree (CART)[19]. Next, we find the optimal -k and +n seconds using a statistical method to set the optimal contextual scene interval around each Point. A video section including a Point is defined as a 'Clip', and the union of all the Clips will be a highlight.

We collect real-time state information from the screen image of E-sports match videos using Optical Character Recognition (OCR) with Convolutional Neural Network (CNN) and pattern matching techniques with Open Source Computer Vision Library (OpenCV)[1]. We chose the 'League of Legends' match videos for this paper, especially 'League of Legends 2018 World Championship'. There are some reasons for choosing the League of Legends match videos:

- The League of Legends is well known as the most popular game in E-sports. Therefore, most E-sports researches are focused on League of Legends. The video data we used in this study, 'League of Legends 2018 World Championship' has 46.7 million viewers per match in average[1], which is about three times of the 2018 World Series viewers(Baseball, 14.3 million per match in average)[2].
- We can collect high-quality data directly. Riot Games, the developer of the League of Legends, provides an Open API to collect the end state information of the official matches. All the match videos and highlights are available on the video platform channel.
- There are a dashboard and some state information in all match videos, and it is good to collect data in a live video in real-time. In addition, there is only one camera in the match, and it provides 60 frames in 1080HD.

[1]League of Legends 2018 World Championship viewers: https://escharts.com/tournaments/lol/worlds-2018

[2]2018 World Series viewers: https://variety.com/2018/tv/news/world-series-ratings-2018-1202994171

2 RELATED WORK AND BACKGROUND

2.1 E-sports Video Highlight Prediction

Researches to predict win-loss of sports matches has been studied. In recent years, machine learning and AI are mainly used [2–7]. These studies built a prediction model of win-loss by the final state information that was the result of the game. There have been studies to predict the real-time wins-loss probability, but the performance of the model was poor. [8–10] We used some of the ideas from related studies and overcame the limitations to create a good performance prediction model. Contrary to the active research on highlights in traditional sports recently [11–17], E-sports does not have much research despite its popularity [18, 19].

Wei-Ta Chu et al. used an event message, which occurred on the screen during a game using Google OCR package [18]. They predicted highlights using an arousal model and an Support Vector Machine (SVM). They simply checked the event messages and predicted highlights. However, precision recorded up to 54% and did not show satisfactory performance. It takes much time to process the match screen to recognize multiple event messages. Since Google OCR package was not optimized for E-sports match videos, there were many image recognition errors as well.

Cheng-Yang Fu et al. predicted highlights through a character level CNN-RNN(Recurrent Neural Networks) model that combined real-time audience chat reactions with visual information of the game itself [19]. The average f1 score was 41.45 for the chat-based model, 70.7 for the video-based model, and 72.4 for the combined model. However, the reason why the video model based on CNN showed the performance of 70.7% is due to the event message, event image, and dashboard that occurred during the match. In addition, these event messages and images can be collected more precisely with just the change of state information in the game. The chat model is only applicable to the case where lots of viewers chat in real-time on a specific platform. It is not applicable if we have a small number of viewers, cannot chat, or if chatting is not active. The model with a CNN-RNN requires much time and computational power because it is too deep.

2.2 Background

We take the terminology used by the creators of highlight in the E-sports industry. Highlight means a summary of the main scenes from the match video. The match video consists of moments for each frame, and some of the important moments are called Points.We tried to mathematically calculate this Point, important moments enough to be included in highlight. Then we defined that the moment that the rate of win-loss probability for each team changed drastically is a Point. State information used as an input is a number representation of information on various dashboards and graphic images existing in the match video. A short video containing a highlight point is called a Clip. Multiple Points can be included in a Clip, but it only each Clip. A Highlight can be generated with a combination of several suitable Clips.

The League of Legends is a multiplayer online battle arena, competes against the Red team and the Blue team in a 5:5 team match. If a team breaks down the opponent's core building called Nexus, the team wins. On the way to Nexus, there are defense building called Towers. Each team does not need to destroy all opponent

RIGHTSLINK

Towers to win, but at least five must be destroyed to attack the Nexus. The player can kill the opposing player. If players kill their opponent players, they can break the tower and nexus without interruption. Gold is the money in the game. Each player can buy weapons and armor to be stronger. Each player can earn gold as a reward for all actions, including killing opponents, towers, and monsters. Dragon is a neutral monster in the match filed. If a team kills the Dragon, they can earn gold and team buffs, the effect a player gets the stronger. It rarely happens. Baron is also a neutral monster, difficult to kill as it gives more gold and stronger effects than a dragon. This event happens very rarely than Dragon kills.

3 OVERVIEW OF PROPOSED METHOD

We try to overcome the limitations of the previous two researches [17, 18] and make better performance of highlight. Our generator can create highlight in real-time, which are comparable to official highlights. It is possible because of optimized video recognition and the highlight Point prediction through the win-loss probability. Optimized means predicting highlights quickly and accurately with the lightest model structure.

In video recognition, we use OpenCV and CNN models. We recognized the state information of the match rather than event messages. The event message is a representation of the outcome of a change in the state of the match. Recognizing only state information, dashboards, and graphic images, it is possible to guarantee stable and better performance than detecting event messages. We tried to identify the dashboard with the simplest CNN structure. Labeling is a priority for learning videos through CNN. However, labeling every frame is unreasonable, with an average of 30 minutes per game. Therefore, we train the CNN model by directly collecting sample data according to the font, size, and color of the dashboard state information. This trained CNN is used as a labeler to label state information in full match videos. In this process, we used openCV for image cropping.

After labeling is complete, we train the CNN to recognize the state information in all target matches. Graphic images outside the dashboard were extracted into data using pattern matching in openCV. Also, for the next step, we recognize the playtime numbers in the target and highlight videos, respectively. We set the playtime as a highlight if a playtime number we recognize in the highlight videos is also present in the target video. That is, in the recognized data set, the playtime is index, the state information is variables, and the label is whether it is highlight or not.

We can calculate win-loss probabilities at every moment in the match using only in-match state information. At some moments, if the win-loss probability changes significantly due to factors affecting win-loss, that moments are clearly the highlight Points of the match. We can predict the win-loss probability of each team by using only state information when the match is over at some Points. Since we collect the end state information in the previous 11,082 official matches, we can construct a standard win-loss classification model. All input data is preprocessed to normalized rate values, rather than to state values for each team because the data we collected is state information at the end of the match. We used MLP with high accuracy (98.4%) of the win-loss classification, so the

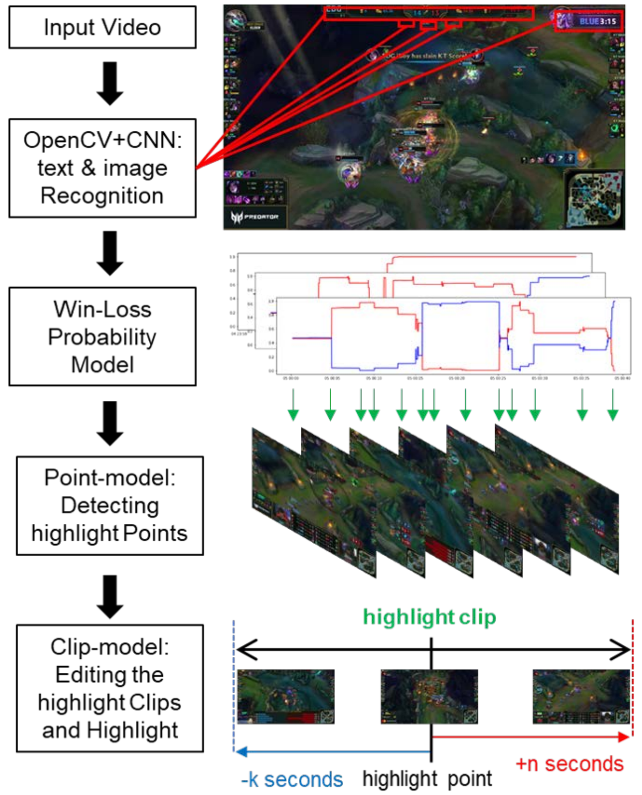

Figure 1: Whole framework of the E-sports video highlight Generator

output value of the last sigmoid node can be used as the win-loss probability.

After we find out the rate of change in the probability of win-loss in every moment, we set a threshold. Points are specified as the moments that are larger than the threshold. To find this, we applied the CART algorithm to calculate the Gini Impurity of whether the moment belongs to the highlight or not, depending on the state information and rate of change in the win-loss probability of the moment. Minimizing the Gini Impurity of the target class in binary classification means that the value best classifies the target class. Then, we used a statistical method to find the optimal Clip interval around Points that can be most similar to the official highlights. Finally, we generate the highlight with the combination of Clips.

Our generator works in the following steps:

- `Step 1. Video Recognition (OpenCV-CNN)` : We collect data in target match videos. Recognition of 'Player kills', 'Tower', and 'Gold' for each team use OpenCV to find outlines and convert images to numbers using CNN model. 'Baron kills' and 'Dragon kills' for each team are recognized by OpenCV pattern matching.
- `Step 2. Win-Loss Probability Model` : It is a pretrained model. We found the variables that affect win-loss in the results data of the last 11,082 matches. And we use this variable to learn with MLP to build a standard win-loss prediction

RIGHTSLINK

model. We set the output value of the last node (sigmoid) to the 'win-loss probability'.

- `Step 3. Highlight Point Model` : We apply this win-loss probability model to the data collected in Step 1 to obtain real-time win-loss probability and state information. We use the rate of change of these values as input to determine the highlight Point.
- `Step 4. Highlight Clip Model` : We create a Clip model that statistically sets the highlight interval using Points determined using the Point model. And compare how the highlight generated by the Clip model predicts the actual formula highlight.

In the following sections, we present the details of each step. Figure 1 shows the whole framework of our approach.

4 VIDEO RECOGNITION (OPENCV-CNN)

4.1 Data

We collected full videos of the 119 matches at the 2018 World Championship on the 'LoL E-sports'[3] and on the 'Onivia League of Legends Highlights' for official highlights[4] on YouTube Channel. All video quality is fixed at 60 frames in 1080HD.

4.2 Model

In order to create the win-loss probability model, we need to obtain the game state information in the middle of matches.

We collected all the state information that can be collected on the match screen: 'Playtime', 'Player kills', 'Tower', 'Gold', 'Dragon kills', and 'Baron kills' for each team. We try to recognize the area for 'Playtime', 'Player kills', 'Tower' and 'Gold' for each team using OpenCV. There are some problems recognizing this information on the match screen. The number images on the match screen are small and keep changing. The image quality deteriorated due to the switching of the screen, and the number images were varied due to the overlapping of the numbers and the background screen. So, we use the approach shown in Figure 2 to recognize the numbers on the screen accurately.

It was difficult to create large amounts of training data from the beginning, so human experts had to preprocess. First, we captured the match screens and cut the number areas from 0 to 9 and label those. We train a CNN model using labeled number images for labeling to unlabeled a large amount of number images. We capture one frame per second in match videos and cut out all the numbers on the screen. To find the number area automatically, we use 'Find Contours' function to find the area of OpenCV. For the numbers found, we use the CNN model to label those images. Through this process, we additionally create 14,696 labeled number images in a match. We use these labeled number images to create an improved CNN. Using the final CNN model, we could perfectly recognize the numbers in all 119 match videos of the 2018 World Championships.

The structure of the CNN models used in this process is the same. Our CNN model accepted images of size 12x16 pixels. A first 2D convolutional layer used neurons with receptive field size $F = 32$,

[3]Full Video Channel (LoL E-sports): https://www.youtube.com/user/LoLChampSeries

[4]Highlight Video Channel (Onivia): https://www.youtube.com/channel/UCPhab209KEicqPJFAk9IZEA

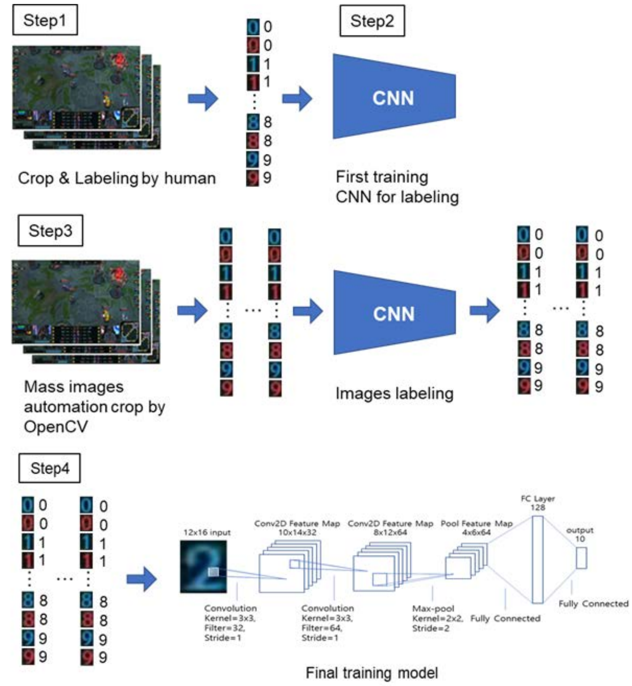

Figure 2: Number recognition in OpenCV-CNN

Figure 3: Image extraction & Pattern matching in OpenCV-CNN

stride $S = 1$ and the number of filters $K = 3$. The output shape of the first 2D convolutional layer is [10x14x32]. Then the output shape of the second 2D convolutional layer (K = 3, F = 64, S = 1) is [8x12x64]. Next, 2D Maxpooling with K = 2 and S = 2, the output shape is [4x6x64]. Finally, a fully-connected 128 layer classifies the number into 10 classes, from 0 to 9. Dropout is done by 0.5.

For image preprocessing, we used the feature of openCV. In the process of resizing each number image to 12x16 pixels, we increase the size of small images. The size of the largest numeric image is 12x16, and the ratio is maintained because all fonts are the same. We also considered interpolation to reduce the large image to medium or small size, but the error rate was high. In the video, two-colored number images existed, and the colors changed little

RIGHTSLINK

by little depending on the screen transition. So we convert the image to a single-channel grayscale and preprocess it to make it look more clearly.

In addition, in order to improve the quality of highlights, we collect Baron and Dragon kills using OpenCV pattern matching. These are special graphical images that are displayed at the bottom of the dashboard and in the upper right of the full screen when the event occurs. In this case, image extraction and pattern matching in openCV was used to recognize whether the graphic image exists or not.

5 WIN-LOSS PROBABILITY MODEL

If the win-loss probability changes significantly at a certain moment, it is clear that a major event has occurred at that moment affecting the win-loss. We assume that that will be a Point in highlight. We have transformed the problem of 'calculating the probability of win at a Point' into 'which team would win if the game ends now?'. We collect state information at the end of the game in the official matches that have been held in the past.

5.1 Data

We collected the result data for the major 11,082 official matches from 2015 to 2018 using the Open API[5] provided by League of Legends developer Riot Games. The major matches included World championships, Mid-Season Invitationals, All-star Events, LCS (North America), LEC (Europe), LCK (Korea), and LPL (China), etc. Because League of Legends is a 5:5 game, there are 10 players in this match data and 184 variables for each team. We have tried various experiments to predict the win-loss. As a result, we chose 6 optimal variables that can predict the win-loss best and be collected on the match screen. The remaining 178 variables have little effect on the rate of win-loss probability, and most of the figures except Baron, Dragon, and End were difficult to recognize on the screen stably.

Table 1: Collected data set using API

Variables	Description	Type
Index	Match Id	String
$X1$	Blue team Gold	Float
$X2$	Blue team Tower	Int
$X3$	Blue team Player kills	Int
$X4$	Red team Gold	Float
$X5$	Red team Tower	Int
$X6$	Red team Player kills	Int
Label	Win	Blue, Red

5.2 Preprocessing & Split

The collected data is the final state (Result) data of matches. We normalize each variable to a relative value between teams to apply to the model that predicts the real-time win-loss probability in the middle of matches.

$$Norm(TS_i) = \frac{TS_i + 1}{\sum_{j=1}^{n}(TS_i + 1)} \qquad (1)$$

[5]Riot games API: https://developer.riotgames.com/

$Norm$ is a relative normalization of each team's state information. TS_i is a team state of the i-th team. n is the total number of teams. We add 1 to each of the numerator and denominator, because the initial value of the TS_i is 0. The input variables of the win-loss probability model are $Norm(Blue\ team\ Gold)$, $Norm(Blue\ team\ Tower)$ and $Norm(Blue\ team\ Player\ Kills)$ as Gold, Tower, Kills.

We used the stratified shuffle split. This cross-validation is a merge of stratified K fold and shuffle split, which returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class[21]. We set the number of splits to 100 and the test size to 30%. The number of Train matches are 7,888 and Test matches are 3,194.

5.3 Model

We find a light and fast model while guaranteeing high classification prediction accuracy for win-loss. The classifier also has to be able to output the proper win-loss probability in the middle of the match.

The model that satisfies these conditions is the MLP, and we train the MLP, as shown in Figure 5. The model outputs the win-loss probabilities of each team. Since we normalized to the relative values of each team, the model has three input variables (Gold, Tower, and Kills). The variables pass through 128 fully connected layers. Each drop out is 0.5. The output values of the sigmoid, which is the last node of the model, is defined as the win-loss probability.

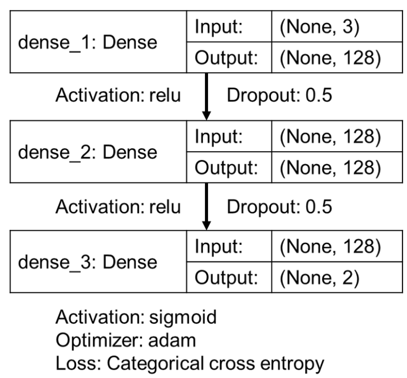

Figure 4: Win-Loss probability model structure

Table 2: Classification performance of win-loss probability model

Accuracy	Precision	Recall	F1
98.4%	98.3%	98.7%	98.5%

5.4 Evaluation

The win-loss probability model classifies win-loss with an accuracy of 98.4%, precision of 98.3%, recall of 98.7%, and F1 score of 98.5%. It is a prediction of the final outcome of the game, based on the end state information. It is a very high score to trust the model. We use this model and calculate the win-loss probability in the middle of matches.

RIGHTSLINK

6 HIGHLIGHT POINT MODEL

We search for the highlight Points by using the rate of change in the real-time win-loss probability. A Moment when the probability value changes drastically is a Point. However, we do not know how much the probability changes so that the moment is the highlight. Therefore, we label highlight at all moments by comparing them with formula highlights and find the optimal threshold using the Gini Index of the CART algorithm.

6.1 Data

The total number of 2018 world championship matches is 119. We split the data into 83 train matches and 36 test matches. By extracting the screen every second of the video, we collect Playtime, Gold, Tower, Player kills, Baron kills, Dragon kills, and the moment the End information of each team using the OpenCV-CNN model. We identified and labeled whether the playtime recognized in the full match video and the highlight match video match each other. Then we calculate the win-loss probability for each team using the win-loss probability model.

Table 3: Dataset after processing in OpenCV-CNN and Win-loss probability model.

Variables	Description	Note
Index1	Match Id	String
Index2	Playtime	mm:ss
Input1	Gold	*Norm(Blue team Gold)*
Input2	Tower	*Norm(Blue team Tower)*
Input3	Kills	*Norm(Blue team Player kills)*
Output1	Win Prob. 1	Probability of Blue team win
Output2	Win Prob. 2	Probability of Red team win
Collected1	Baron	*Norm(Blue team Baron kills)*
Collected2	Dragon	*Norm(Blue team Dragon kills)*
Collected3	End	End or Not
Lable	Highlight	Highlight or Not

6.2 Preprocessing

To find the highlight Points, we preprocess the win-loss probability for each team using $\Delta(x_t)$, an absolute time difference of x_t. x_t is a value x at Playtime t. We set $x_0 = x_1$ because there is no value change after 1 second of starting.

$$\Delta(x_t) = |x_{t-1} - x_t| \tag{2}$$

Δ*(Win-loss probability)* is the sum of Δ*(Probability of Blue team win)* and Δ*(Probability of Red team win)*. Labels are set to 1 if a playtime exists in the Clip of the official highlight, or 0 if it does not exist.

6.3 Model

We find the highlight Point based on Δ*(Win-loss probability)*, the sum of the rate of change in win-loss probability for each team. For convenience, we will call Δ*(Win-loss probability)* with '*P1*'. Then we use the Gini Impurity of the CART algorithm to determine a threshold of P1.

Set P contains values from j classes, '$P > k$' is a subset of P, and k is a real number. So $P > k$ is also containing values from j classes. p_i is a fraction of items with class i in $P > k$.

$$Gini(P > k) = \sum_{i=1}^{j} p_i(1 - p_i) \tag{3}$$

$Gini(P > k)$ is an Gini Impurity which is the sum of $p_i(1 - p_i)$ values for all class i in $P > k$. We find a k that minimizes the value of $Gini(P > k)$. It means that the highlight class is best split when $P > k$, regardless of no-highlight class. We define this process as a model called *Point(P)*. *Point(P)* is expressed as follows according to the value of k which minimizes $Gini(P > k)$:

$$Point(P) = \begin{cases} HighlightPoint & \text{if } P > k \\ Not & otherwise \end{cases} \tag{4}$$

6.4 Result

We found a value of k that minimizes the value of $Gini(P1 > k)$. We confirmed that it has the lowest the Gini Impurity value, 0.292 when $P1 > 0.005$.

$$Gini(P1 > 0.005) = 0.292 \tag{5}$$

We can specify playtime, which satisfy $P1 > 0.005$ as the highlight Point. As a results of Point model, 82.02% of *Point(P1)* when P1>0.005 belong to the test set of official highlights. Regardless of the *Gini* values, we can newly define the models that specify the top 1% of P1 and $P1 > 0.005$ as *Point (P2)*, and top 5 of P1 and $P1 > 0.005$ as *Point(P3)*. *Point(P2)* is 68.1%, and *Point(P3)* is 53.3% belong to the test set of official highlights. Likewise, we can make various models by adjusting the threshold. In addition, we applied Baron, Dragon and End which are very rare and the biggest events in the match. These are 0 threshold, which means that if any event occurs during the match, it would be a highlight.

$$Point(P1) = \begin{cases} Chosen & \text{if } P1 > 0.005 \\ Not\ chosen & otherwise \end{cases} \tag{6}$$

$$Point(P2) = \begin{cases} Chosen & \text{if } Top\ 5\ of\ P1 \cap P1 > 0.005 \\ Not\ chosen & otherwise \end{cases} \tag{7}$$

$$Point(P3) = \begin{cases} Chosen & \text{if } Top\ 1\%\ of\ P1 \cap P1 > 0.005 \\ Not\ chosen & otherwise \end{cases} \tag{8}$$

$$Barons = \begin{cases} Chosen & \text{if } \Delta(Baron) > 0 \\ Not\ chosen & otherwise \end{cases} \tag{9}$$

$$Dragons = \begin{cases} Chosen & \text{if } \Delta(Dragon) > 0 \\ Not\ chosen & otherwise \end{cases} \tag{10}$$

$$Ends = \begin{cases} Chosen & \text{if } \Delta(End) > 0 \\ Not\ chosen & otherwise \end{cases} \tag{11}$$

RIGHTSLINK
Copyright Clearance Center

7 HIGHLIGHT CLIP MODEL

A highlight is a set of Clips that are a short video section around a Point. If Point model selects the appropriate Points, then we determine the intervals around the Points. We have highlight Points specified at *Point(P1)*, *Point(P2)* and *Point(P3)*. We consider whether we can use any events we can detect. We extract three major events that can be an additional highlight Points. These are Barons, Dragons, and Ends, regardless of teams.

We investigate how many included these events are in official highlights. As we would expect, those three variables are unconditionally included in the highlight because they are the biggest events that occur once or twice in a game. In comparison with our *Point(P1)*, Barons are belonged to 7.5%, Dragons to 13.7% and Ends to 2%. We can combine these events as additional Points to increase the accuracy of the model.

Table 4: Event comparison with Official highlights and *Point(P1)*

Event	# of Events	Official Highlights	*Point(P1)*
Barons	160	160	12
Dragons	248	248	34
Ends	119	119	3

7.1 Data

The Clip model's input is the highlight Points from the output of Point models. We mainly consider *Point(P1)*, *Point(P2)*, and *Point(P3)*. We additionally use detected 3 events, Barons, Dragons, and Ends. All events that can be collected through the OpenCV-CNN model are considered.

7.2 Model

The interval of each highlight Point is estimated using statistical techniques, compared to official highlights. We predict the lengths of before and after each highlight Point. Analysis of Clips of the official highlights, Towers was not in the rules. 37% Towers were included because another event occurred simultaneously with Towers. Therefore, Towers are not considered separately but considered through *Point(P1, P2, P3)*. Table 5 shows the average and standard deviation of Clips for each Point that exist in official highlights. In the actual model, We used the mean but cut off the decimal point. Thus, Point (P1), Point (P2), Point (P3) is (-12, 7) seconds, Barons is (-11, 4) seconds, Dragons is (-3, 2) seconds, Ends is (-27, 0) seconds were applied. We named this method C1, C2, C3, C4, C5, C6 respectively.

We use a combination of various highlight Points to predict the Clips. The Clip model is defined as follows:

$$Clip(C1, \cdots, CN) = Clip(C1) \cup \cdots \cup Clip(CN) \tag{12}$$

We generate several versions of highlights by combining Clips from C1 to C6 in Table 6. The final versions we selected were a total of 6, as Win-loss, Top 5, Top 1%, Win-loss + event, Top 5 + event, and Top 1% + event. We will call this from M1 to M6.

Table 5: Mean (standard deviation) time interval of Clips containing highlight Points

Method	Point	-Second	+Second
C1	Point(P1)	-12.04(2.70)	7.23(1.78)
C2	Point(P2)	-12.04(2.70)	7.23(1.78)
C3	Point(P3)	-12.04(2.70)	7.23(1.78)
C4	Barons	-11.67(7.47)	4.58(2.47)
C5	Dragons	-3.61(3.45)	2.04(1.55)
C6	Ends	-27.33(5.91)	0.00(0.00)

Table 6: Highlight Clip models

Method	Name	Clip
M1	Win-loss model	$Clip(C1)$
M2	Top 5 model	$Clip(C2)$
M3	Top 1% model	$Clip(C3)$
M4	Win-loss + event	$Clip(C1, C4, C5, C6)$
M5	Top 5 + event	$Clip(C2, C4, C5, C6)$
M6	Top 1% + event	$Clip(C3, C4, C5, C6)$

Table 7: Classification performance of win-loss probability model (%)

Method	Accuracy	Precision	Recall	F1
M1	88.2	73.0	64.6	68.5
M2	80.1	50.2	11.5	18.8
M3	83.0	62.5	36.4	46.0
M4	**89.9**	**74.7**	**74.4**	**74.5**
M5	82.5	65.6	25.9	37.2
M6	85.0	67.4	47.9	56.0

8 HIGHLIGHT EVALUATION

8.1 Comparison with Official Highlights

We evaluated how well the highlights we generated predict the official highlights. The total number of 2018 world championship matches is 119. The number of Train matches is 83 and test matches is 36. In the case of M1, M2, and M3, the highlights are generated by designating an appropriate threshold value for the variation of the real-time win-loss probability. In the case of M4, M5, M6, additional events such as Barons, Dragons, and End Points are considered. In the case of the M4, it is the model most similar to the official highlight video, with 89.9% accuracy. All the evaluation results of these models are shown in Table 7.

8.2 Human Evaluation

We compare M4 to the official highlights by human evaluation. We request two tasks to those who enjoy League of Legends through an experimental booth and an online questionnaire on the League of Legends community.

The first task is choosing better highlights. We show the M4 model highlights and the official highlights at the same time and request to choose a better highlight. We show the 100 vs. FNC

RIGHTSLINK

match of world group stage day 2. The two highlights are almost similar in length. If 50% of the surveys come out, it means that our highlight are indistinguishable from official highlights. As a result of 128 people, 84 people (65.62%) answered that the M4 is a better highlight than the official highlight. They said "These highlights are more natural.", "These highlights helped me to better understand the flow of the game.", "There were many scenes that were not important to other highlights.", etc.

The second task is to compare exclusive Clips. We exclude Clips included in both highlights from each highlight. We randomly choose 200 Clips, which are included in only our highlights and another 200 Clips, which are included in only the official highlights. We show randomly chosen 20 Clips from the 400 Clips to each evaluator. We ask 20 people to rate the importance of Clips on a 5-Point scale. The average Point of Clips included in our highlights is 4.02, and that of the official highlights is 2.63. The Cronbach's alpha value for this survey is 0.749, which is acceptable. It is estimated that the Clips that exist only in our highlights are more important than those in the official highlights. In other words, people responded that our highlights are better than the official highlights.

In addition, we analyzed the Clips that existed only in ours, and over 82.3% of the Clips with average scores higher than 4 Points belonged to the highlights made with M5 and M6. We can argue that the highlight made with the M5 and M6 effectively scaled the length of the highlighted video, though it might outperform the M4. Highlights made with M5 and M6 are shorter than M4 by an average of 72 seconds and 43 seconds, respectively.

9 CONCLUSION

We propose a new method of generating highlight in real-time in E-sports match videos, using the win-loss probability model. Our highlight generator is based on the idea that moments with large rates of change in win-loss probability are highlight points. Since all the data is recognized and processed immediately, the operation speed of the model is real-time. The highlight generated, similar to the most official highlights, reached 89.9% accuracy. The accuracy of this was also measured indirectly by people's evaluation. Also, our highlight generators can easily be created in various versions, which is enough by adjusting the threshold. We made our highlight by comparing it with official highlights, but without these constraints, it is easy to make more various versions of highlights simply by adjusting some of the conditions and parameter values. Since we can prioritize highlight Points, it is possible to select only 1% of the important Clips in the matches. We can also combine Top 5 Clips to create the new highlights.

In the next step, we are going to study how to predict fast, accurate, and customized highlights by applying our approach to various sports. If we design the optimal win-loss probability model for each sport well, we can apply this research methodology.

ACKNOWLEDGMENTS

This work was supported by Institute of Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00421, AI Graduate School Support Program)

REFERENCES

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
[2] Mohammadreza Yamaghani Milad Keshtkar Langaroudi. Sports result prediction based on machine learning and computational intelligence approaches: A survey. *Journal of Advances in Computer Engineering and Technology*, 5(1):27–36, 2019.
[3] Leandro A.Neves Luciano V.Lima Edna L.Flores Marcelo Z.do Nascimentod Rodrigo G.Martins, Alessandro S.Martins. Exploring polynomial classifier to predict match results in football championships. *Expert Systems with Applications*, 83(15):79–93, 2017.
[4] D.Miljković, L.Gajić, A.Kova cević, and Z.Konjović. The use of data mining for basketball matches outcomes prediction. *IEEE 8th International Symposium on Intelligent Systems and Informatics*, pages 309–312, 2010.
[5] Nick Sephton Florian Block Anders Drachen Peter Cowling Victoria Hodge, Sam Devlin. Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games. 2017.
[6] Chen W. Qin R. Nummenmaa T. Nummenmaa J. Lan X., Duan L. A player behavior model for predicting win-loss outcome in moba games. *Advanced Data Mining and Applications*, 11323:474–488, 2018.
[7] Ku and J. Kim. Development of game indicators and winning forecasting models with game data. *Journal of the Korean Data and Information Science Society*, 28(2):237–250, 2017.
[8] Yu-Heng Lei Yifan Yang, Tian Qin. Real-time esports match result prediction. 2016.
[9] Korolev S. Yashkov D. Neklyudov K. Semenov A., Romov P. Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. *Analysis of Images, Social Networks and Texts*, 661:26–37, 2017.
[10] Nick Sephton Florian Block Anders Drachen Peter Cowling Victoria Hodge, Sam Devlin. Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games. 2017.
[11] M. Merler, D. Joshi, Q. Nguyen, S. Hammer, J. Kent, J. R. Smith, and R. S. Feris. Automatic curation of golf highlights using multimodal excitement features. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 57–65, 2017.
[12] M. Merler, K. C. Mac, D. Joshi, Q. Nguyen, S. Hammer, J. Kent, J. Xiong, M. N. Do, J. R. Smith, and R. S. Feris. Automatic curation of sports highlights using multimodal excitement features. *IEEE Transactions on Multimedia*, 21(5):1147–1160, 2019.
[13] Tom Decroos, Vladimir Dzyuba, Jan Van Haaren, and Jesse Davis. Predicting soccer highlights from spatio-temporal match event streams. *AAAI'17*, pages 1302–1308, 2017.
[14] Pushkar Shukla, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. Automatic cricket highlight generation using event-driven and excitement-based features. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
[15] H. Kim, T. Mei, H. Byun, and T. Yao. Exploiting web images for video highlight detection with triplet deep ranking. *IEEE Transactions on Multimedia*, 20(9):2415–2426, Sep. 2018.
[16] Ali Javed, Aun Irtaza, Yasmeen Khaliq, Hafiz Malik, and Muhammad Tariq Mahmood. Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine. *Applied Intelligence*, 49(8):2899–2917, 2019.
[17] Y. Jiao, Z. Li, S. Huang, X. Yang, B. Liu, and T. Zhang. Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE Transactions on Multimedia*, 20(10):2693–2705, Oct 2018.
[18] Wei-Ta Chu and Yung-Chieh Chou. Event detection and highlight detection of broadcasted game videos. *HCMC '15*, pages 1–8, 2015.
[19] Cheng-Yang Fu, Joon Lee, Mohit Bansal, and Alexand er C. Berg. Video Highlight Prediction Using Audience Chat Reactions. *EMNLP*, pages 972–978, 2017.
[20] R. Olshen L. Breiman, J. Friedman and C. Stone. Classification and Regression Trees. *Wadsworth Int. Group*, 1984.
[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

RIGHTSLINK