Application of Frame Selection For Binary Video Classification

Application of Frame Selection for Binary Video Classification

By Zichao Zhao,

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment of the Requirements for the Degree M.A.Sc.

McMaster
M.A.Sc. (2019)
Hamilton, Ontario (ECE)

TITLE: Application of Frame Selection for Binary Video Classification
AUTHOR: Zichao Zhao (McMaster)
SUPERVISOR: Dr. Jun Chen
NUMBER OF PAGES: vii, 50

Abstract

This thesis presents frame selection based on genetic algorithm and Euclidean distance and its exploitation on binary video classification. The frame selection implementation provides a fast enhancement of classification results.

Parallel frame selection methods are put up in comparison and a series of performance enhancement models are proposed to scrutinize the relationship with frame selection and binary video classification.

Other approaches might also be valid to be exploited in the frame selection baseline in order to improve classification results. We could learn semantic meanings with the assistance of natural language processing, combine audio with pixel-level information to form a fusion model, or directly manage learning methods on video level.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude of thanks to people who have helped me a lot to make the completion of this thesis possible. First and foremost, I would like to express my sincerest appreciation to my supervisor Dr. Jun Chen, for his kindness, great patience and wise advice. In addition, I am also very grateful to Dr. Dongmei Zhao and Dr. Jiankang Zhang for serving on my thesis defense committee. I would like to appreciate all the colleagues in my lab as well as staff members in the department, in particular Cheryl, who provided kind assistance in my period in McMaster University.

In the meantime, I would also like to express my appreciation to Mrs. Liza Wood, Mr. Luke Liu and Mr. David Wang for their assistance during my internship.

Contents

Abstract iii

Acknowledgements iv

1 Introduction 1

2 Methodologies 3
2.1 Generation of Chromosome Population . . . 3
2.2 Frame Dictionary Formation . . . 4
2.3 Genetic Algorithm Application During Selection . . . 6
2.3.1 Encoding . . . 6
2.3.2 Fitness Function (Inside Key Frame Groups) . . . 6
2.3.3 Mutation and Crossover . . . 7
2.4 Result Evaluation Metrics . . . 9
2.4.1 Two class Recall-Precision . . . 9
2.4.2 ROC curve . . . 10
2.4.3 Accuracy . . . 12
2.4.4 Recall . . . 12
2.4.5 Precision . . . 12
2.4.6 F1 score . . . 13

3 Experimental Results, Analysis and Improvements 14
3.1 Experiment Results . . . 14
3.1.1 Classification results on test set . . . 15
3.1.2 Representation of video classification . . . 16
3.2 Analysis . . . 17
3.2.1 Analysis on Test Set Result . . . 17
3.3 Improvements . . . 22
3.3.1 Image Classification Level Improvements . . . 23
3.3.2 Decision Making Level Improvements . . . 27
3.3.3 Frame Selection Level Improvements . . . 32
3.4 Conclusion . . . 36

4 Future Works 37
4.1 Pre-classification and NLP Level Improvement . . . 37
4.1.1 Transfer From PDF Element Classification . . . 38

4.1.2 OCR 41
4.1.3 Semantic Analysis 41
4.2 Multi-Feature Fusion 42
4.3 Video Level Training Approaches 44
4.3.1 Deep Feature Flow 44
4.3.2 Adaptive Scan 46

Bibliography **49**

List of Figures

3.1 Selected frames in the hand video 17
3.2 Wrong predicted frames in a mechanical operation video 19
3.3 Correctly predicted frames in a mechanical operation video 19
3.4 Wrong predictions from normal boxing, swimming scenes 20
3.5 Wrong predicted frames in naked skin video 21
3.6 Correctly predicted frames in a naked skin video 21
3.7 Examples of frames of animals being predicted as porn 21
3.8 Average Precision and AUC for ensembles 25
3.9 Best Average Precision for the ensembles 27
3.10 Skin area detection for a wrestling scene 34
3.11 Limitation of skin area detection 35

4.1 Some frames selected contain direct message 37
4.2 Other cases when language information was involved 38
4.3 Figure predicted by the PDF classifier 39
4.4 "words-related" clips predicted by the PDF classifier 40

Chapter 1

Introduction

In order for screening unwelcome information in online resources accessible to underage adolescences, classifiers identifying whether images contain pernicious contents were constructed with the thrive of deep learning in computer vision area. It introduces *ia classifier*, an image classifier available upon request, provides verification towards pornography scenes.

To realize experiments and validation, we made use of **Pornography-2k** dataset, a new 65 challenging pornographic benchmark that includes 2000 web videos and 140 hours of video footage.

Comparing to image classification, video classification could be restricted to multiple factors. In practice, video classification resources still cannot compete with existing image datasets in terms of scope and category since they are significantly more complicated to be annotated or stored. In order to be accepted as inputs for deep neural network such as CNN, images were preprocessed into fixed size. However, videos diverse widely from resolution, format as well as duration. These make preprocessing extremely onerous and unpredictable.

Videos are composed of millions of continuous frames. To balance predictive accuracy and resource availability, we implemented a combination of image prediction scores to represent the video classification result. Thus, we put forward a Euclidean distance based frame selection method to collect the most representative scenes in videos.

Chapter 2

Methodologies

2.1 Generation of Chromosome Population

Accessing a video to be processed, we collected its duration length N in the unit of number of frames. We randomly generated an $n \times m$ matrix of frame indices, where m indicates the number of key frames to be selected. The m-dimensional vector represents a combination with a series of indices, which would be utilized to calculate inner average Euclidean distance for the later steps. Each vector has a list of increasing indices to be in accordance with video frame access order.

Since the frame numbers are set from *0* to *N-1* by default, it must be guaranteed that all the frames to access are potentially possible to be selected. In the meantime, this is obscure because different libraries such as **cv2** or **lintel** might have discrepancies in reading frames. Thus, it was critical to make sure the duration boundaries would never be breached. We generated different m-dimensional vectors in parallel for n times where n equals to:

```
len(range(start_idx, N+1, m))
```

start_idx could be customized by user. This was valid because the longer the video the more groups of combination would be analyzed. From the larger base, we could collect a more accurate result with the largest possible average Euclidean distance.

We obtained an $n \times m$ matrix following this procedure, with each inner element being an index number that is accessible in the video. It physically represents n pieces of m-dimensional segment combination.

The number of key frames to be selected was set to 10 in our experiment, as most cases in **Pornography-2k** dataset have large duration so that the collection of 10 frames could be guaranteed, and larger key frame numbers require more times for processing. For special cases with very small video clips, we make sure the key frames selected m to be $N \bmod m$ so that the upper bound would not be breached.

2.2 Frame Dictionary Formation

When using **opencv** library for accessing frame information, a dictionary was generated for each video with the keys referring to all the frame indices existed in the video, while the values being **numpy** matrices representation of corresponding frames. Each frame was resized to 256×256 with a bicubic interpolation over 4×4 pixel neighborhood and converted from original RGB to greyscale.

It was meaningful to point out that since we generated random indices from previous step, we could read merely the necessary frames instead of all of them into represented dictionary. In practice, however, when we define a **VideoCapture** object in **cv2** and implement

```
set(cv2.CAP_PROP_POS_FRAMES, i)
```

method to access specified frames (i indicates the specific index generated previously), overall running time was longer in such iterations than reading all the frames in. This was due to an optimization problem affluently existed in **opencv** frame reading module. To solve this problem, we also introduced a **lintel** based frame loading method. **lintel** is a video decoding module which provides fast and simple implementation in Python scripts. The method particularly decoded a set of frames from file stream object into a 4-d **numpy** array, which perfectly fitted our realization of giving a list of indices, as well as the objective of acquiring matrices in sequence accordingly. Another advantage was that original dimensionality information was not indispensable as the width and height could be determined by an underlining library called **libavcodec**. The simplicity of creating objects and the accordance of parameter definition provided with more efficiency and guaranteed quicker data access. The first dimension length of this output 4-d tuple was in accordance with the customized key frame indices to be selected, and each frame matrix was also converted to 256×256 greyscale, to keep track with the **opencv** method. In this way, we could abandon the frames whose indices were not randomly chosen in the previous step, and improved efficiency.

2.3 Genetic Algorithm Application During Selection

2.3.1 Encoding

This part illustrates the encoding mechanism, and its relation with chromosome generation [1]. As mentioned in 2.1, the m-dimensional vector indicates a combination of key frame indices in the video. This m-dimensional vector could be interpreted as a chromosome, which is a single unit of the $n \times m$ matrix. Random generation built the foundation of genetic algorithm, allowing a unity of indices to be selected eventually. Each element in the matrix might be modified according to whether a randomly assigned decimal is above the crossover probability set by user manually.

2.3.2 Fitness Function (Inside Key Frame Groups)

Since the indices were sorted in ascending orders, the **numpy** array representation collected above was in temporally sequential order. This guaranteed robustness when calculating average Euclidean distance. Consecutive frames inevitably share similar contents with larger possibility than discrete frames with considerable time differences. To clarify, suppose 3 key frames a-b-c were selected from a specific video clip and they were organized in sequential order. Under this circumstance we can access frame dictionary formed from previous step, and collect values: **numpy** array corresponding to a and b, and assign Euclidean distance between them to a temporary summation variable. For the next step we could directly add distance between b and c to update total distance.

If they were extracted out of ordered, such as a-c-b, this three-moment clip would have a larger overall Euclidean distance. Because distance between a and c includes differences between b and c so multiple distances would be calculated in such situations. This accessing mechanism also allows each frame to be asked only for once. This property should be followed undoubtedly during iterations through the population, when implementing summation of Euclidean distance.

During iterations, each index should be interpreted as key to the frame dictionary, distance between neighboring frames was calculated by norm-2 distance:

```
im1 = frame_dict[ind1]

im2 = frame_dict[ind2]

ed_sum += cv2.norm(im1, im2, cv2.NORM_L2)
```

Final result was represented by average summation. Recall that there were m key frames to be chosen. The single group should have

```
ed_sum / (m-1)
```

as representation of average Euclidean distance.

2.3.3 Mutation and Crossover

For each iteration, 3 rows was randomly selected from the $n \times m$ population P. Let us denote this $3 \times m$ matrix by R. A mutation value mv was calculated via

```
int(P[j][i] + F*(R[1][i] - R[2][i]))
```

where i iterated through *0* to m (total key frame to be selected) - 1. j was defined manually by user, which was set to iterate inside the first dimension of P. F was set as 0.9 in our experiment. If mv is lower than *0* or over N (the max frame number in a video) - 1, the index would be out of control. Thus, we limited the mv as upper or lower limit. Other than that, mv was updated with the above equation.

Within key frame numbers, 10 mv values were generated. Through each iteration, these 10 indices formed a new Mutation Vector.

The crossover probability C_r was set to 0.6. We randomly generated a decimal from Uniform Distribution(0, 1). If this number is lower than C_r, a mutation value corresponding to current iteration index i would be picked, otherwise the algorithm would pick the original i_{th} item. In this way, we generated a new Trail Vector.

The Trail Vector was obviously an offspring comparing to its original counterpart. The parent and offspring were bring together to calculate the fitness function. The one with higher average Euclidean distance was picked, thus updating the original corresponding chromosome.

The above process updated the $n \times m$ population. Each chromosome inside possessed a possible largest average Euclidean distance. Finally, an optimal combination would be chosen from these n chromosomes, to represent our key frame indices.

2.4 Result Evaluation Metrics

When passing an image (frame) into image classifier, it would provide predictions of related video with respect to its possibility of being non-porn and porn. These predictions were represented in decimals. We could also generate binary predictions if certain threshold is applied. For example, if the prediction from *ia classifier* of a video is above threshold *t*, we would assign that frame with a binary prediction *1*, whereas this binary number being *0* if less than *t*. The formal figure is useful for evaluating overall performance of the classifier, thus indirectly reflect the effect of key frame selection on image and video classification. The binary prediction works perfectly with metrics defined in **sklearn** Python package, we can calculate *accuracy_score*, *f1_score*, *recall_score* and *precision_score* respectively. It also offers intuitive comparison vertically between various metrics of deciding the video classification representation, thus providing ideas on metric improvements and modifications and should be critical for decision making.

2.4.1 Two class Recall-Precision

In **sklearn** module, both *average_precision_score* and *precision_recall_curve* works with binary classification, which fits our objective in this paper. Average precision (AP) from prediction scores is defined by:

$$\mathrm{AP} = \sum_n (R_n - R_{n-1}) P_n \tag{2.1}$$

where P_n and R_n refer to precision and recall at the n_{th} threshold. *precision_recall_curve* computes *precision_recall* pairs for different probability thresholds. Putting these pairs into coordinates provides us with a two-class Precision-Recall curve, the area under the curve provides us with a percentage of how well our prediction might be.

2.4.2 ROC curve

Receiver operating characteristic (ROC) curve displays diagnostic ability of binary classification system when variations exist in its discrimination threshold. The vertical axis values of the points on ROC curve represent true positive rate whereas the horizontal axis values represent false positive. When a different binary threshold is put forward, true positive rate and false positive rate will vary accordingly, and thus forming the entire curve. Area under curve (AUC) quantizes ROC curve, it is a critical standard for evaluating prediction performance, which indicates the integral area of the ROC curve. Larger true positive as well as larger false positive guarantees a better performance. In conclusion, prediction with larger AUC excels.

Notation

- True Positives (TP) - First, a True Positive must be a video labelled as porn. After generating decimal prediction for each frame selected by image classifier, a criterion or algorithm is implemented to assign this video with a binary video prediction. The binary prediction matches its label, which is positive, then this case is a True Positive. In a word, it symbolizes the case of predicting a porn as positive correctly.

- True Negatives (TN) - First, a True Negative must be a video labelled as non-porn. After generating decimal prediction for each frame selected by image classifier, a criterion or algorithm is implemented to assign this video with a binary video prediction. The binary prediction matches its label, which is negative, then this case is a True Negative. In a word, it symbolizes the case of predicting a non-porn as negative correctly.

- False Positives (FP) – First, a False Positive must be a video labelled as non-porn. After generating decimal prediction for each frame selected by image classifier, a criterion or algorithm is implemented to assign this video with a binary video prediction.

  The binary prediction is inconsistent with its label, which is positive, then this case is a False Positive. In a word, it symbolizes the case of predicting a non-porn as positive incorrectly.

- False Negatives (FN) – First, a False Negative must be a video labelled as porn. After generating decimal prediction for each frame selected by image classifier, a criterion or algorithm is implemented to assign this video with a binary video prediction. The binary prediction is inconsistent with its label, which is negative, then this case is a False Negative. In a word, it symbolizes the case of predicting a porn as non-porn incorrectly.

2.4.3 Accuracy

Accuracy can provide with most direct and intuitive evaluation for the performance of a model. It scrutinizes the percentage of all the correctly predicted cases in all the experiment sample. Of course, we would like to maximize accuracy for models. But for datasets that are asymmetric, where False Positives and False Negatives differ to large extent, accuracy becomes invalid to evaluate our model, so new metrics are required to be introduced.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{2.2}$$

2.4.4 Recall

Recall, or sensitivity,looks through all the positive cases in a dataset, and calculates the percentage of True Positives inside. Specifically in our experiment, it observes the ratio of correctly predicted porn to all the porn videos in **Pornography-2k**.

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

2.4.5 Precision

Precision displays the percentage of correctly predicted positive pornography cases to the total cases being predicted as porn.

In reality, pornography judgement requires us to identify positive cases as many as possible. Thus, the cost of predicting a porn as non-porn is more unacceptable than predicating a non-porn as porn. Of course we would like higher performance on each metric, but in comparison, we have a preference of maintaining recall on a higher level, since the essence of the task is to lower False Negatives on largest scale. If the cost of FPs and FNs differ on a large scale, precision and recall would be a better duo for evaluation.

$$precision = \frac{TP}{TP + FP} \tag{2.4}$$

2.4.6 F1 score

F1 Score is a combination of Precision and Recall, a weighted sum. Therefore,it considers both FPs and FNs. It is intuitively trickier but more robust than accuracy, especially when positives and negatives have an inclination of imbalance.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.5}$$

Chapter 3

Experimental Results, Analysis and Improvements

3.1 Experiment Results

The **Pornography-2k** dataset is composed of pornography videos as well as non-pornography videos with a proportion of 1:1. Out of experimental usage, it was randomly divided into training set, test set and validation set with the ratio of 2:1:1. It is necessary to point out that before Euclidean distance based key frame selection was introduced, a basic random frame skipping strategy was implemented to obtain frames. By default, the frame number to be selected was restricted to 15-20, and each new frame selected was at least 5 seconds later than previous one. For detailed calculation, time can be converted to frame numbers by multiplication with frame rate(frame per second). On the other hand, for video clip whose duration was not long enough for such configuration, number of seconds to skip was converted to number of frames to skip, allowing all the cases to be accessed without content exceeding.

Taking advantage of image classifiers, the score was predictive representation to a single frame. We intuitively combine frame results from max, min, mean and median respectively, to embody the final prediction for a single video. No matter random (15 or 20 frames selected in most cases, let alone specifications with very short video clips) or Euclidean distance (10 frames selected in nearly all cases in **Pornography-2k** dataset) based frame selection method was utilized, max had an overall better score in terms of average precision score as well as area under curve.

3.1.1 Classification results on test set

The following results combining *ia classifier* with random selection and Euclidean based key frame selection was get from **Pornography-2k** test set.

Method	AP	AUC
Random (max)	0.9607	0.9493
Random (mean)	0.9529	0.9474
Random (min)	0.8625	0.8473
Random (median)	0.9559	0.9492
Euclidean (max)	0.9683	0.9634
Euclidean (mean)	0.9501	0.9509
Euclidean (min)	0.8343	0.8176
Euclidean (median)	0.9560	0.9536

For easier comparison, Euclidean based method with maximum will be defined as **Baseline** from here.

3.1.2 Representation of video classification

We can find out a pattern comparing random selection with Euclidean distance based method.

First, we need to trade off between two situations: one is, predicting a normal stream as pernicious while the other is describing harmful scenes as harmless, they correspond with False Positive and False Negative with respectively. No matter which metric is implemented, the result is always brimming with former cases of different reasons, detailed situation will be listed in later part. For example, top naked relevant sites such as summer pool might have a large probability of being decided as porn scene depending on exposed skin area. However, if masturbation or violent sex scenes are not able to be identified, then the main purpose of protecting kids from not discernible online resources would be sacrificed.

Initially analyzing the results above, since minimum metric yielded lowest scores, it would definitely be eliminated from the final metric. Gathering with the trade off analysis from above, this is also ideal and reasonable since we always would like to scrutinize the highest scores predicted by image classifiers.

Set aside minimum, Euclidean gave an overall better performance on AP and AUC. The fact that both methods yielded best performance when using maximum substantiated our interest in the maximum score given by image classifier among the selected frames. This strategy would also maximize our possibility of finding out positive cases. Although this could lead to rise in the number of False Positives, and in turn sacrifice

FIGURE 3.1: Selected frames in the hand video

precision, True Positives can be identified with a high probability. Thus, precision won't be sacrificed for a large basis. Also to alleviate this problem, we could also purpose noise examination to achieve better score.

3.2 Analysis

3.2.1 Analysis on Test Set Result

From here, we need to analyze all the possible reasons lead to wrong predictions when Euclidean distance based key frame selection was applied to video classification baseline. Threshold was set to 0.5, in consistent with previous binary classification assignment. Maximum is also the chosen metric for this evaluation.

Potential problems existed:

1. Image Classifier we utilized in the experiment misclassified multiple images in a

single video, which indicates that the image classifier might be problematic recognizing certain pattern of same scenario.

This situation mostly yielded False Positives affluently. For example, in a very short video clip that showed movement of a hand on the beach, 4 out of 10 frames were given a prediction score above 0.5 (FIGURE 4.1). This video has content relatively deceptive for an image classifier, since the connection part between finger and palm might be resemble with bottom.

On another occasion, a mechanic that lies under a car was repeatedly predicted as porn. Certain movements or postures might also be tricky.

2. *ia classifier* misclassified one of the ten frames, so that the max score selected was above 0.5. In general, this equals noise and could only yield False Positives. For example, animal-related clips have an inclination of being mistakenly predicted.

3. Naked skin problem. This could be ascribed to problem 1, but there was a vital difference. Each problematic case categorized as problem 1 had a particular character, which could not be summarized as a single class.

   For example, the finger video contained pixels which were confused for classifiers, the mechanical video we mentioned was a problem with posture. Thus, problem 1 was more subtle to be addressed. However, problem 3 was the most vital pattern exist in False Positives. The larger the naked skin areas exposed to screen, the larger the possibility that a video being classified as porn. In our test set with 500 cases (half porn, half regular), 20 out of 28 False Positives had this naked skin

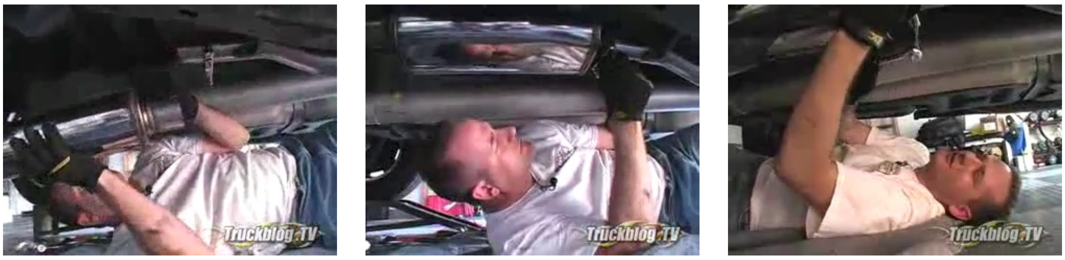


FIGURE 3.2: Wrong predicted frames in a mechanical operation video

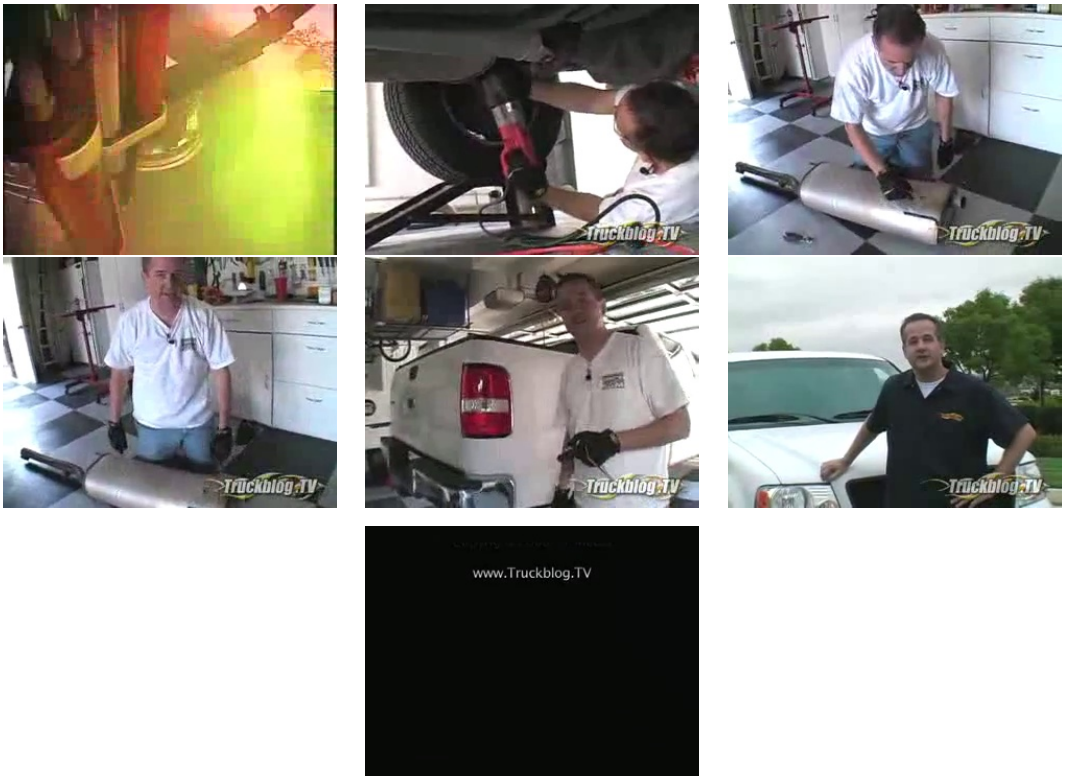


FIGURE 3.3: Correctly predicted frames in a mechanical operation video

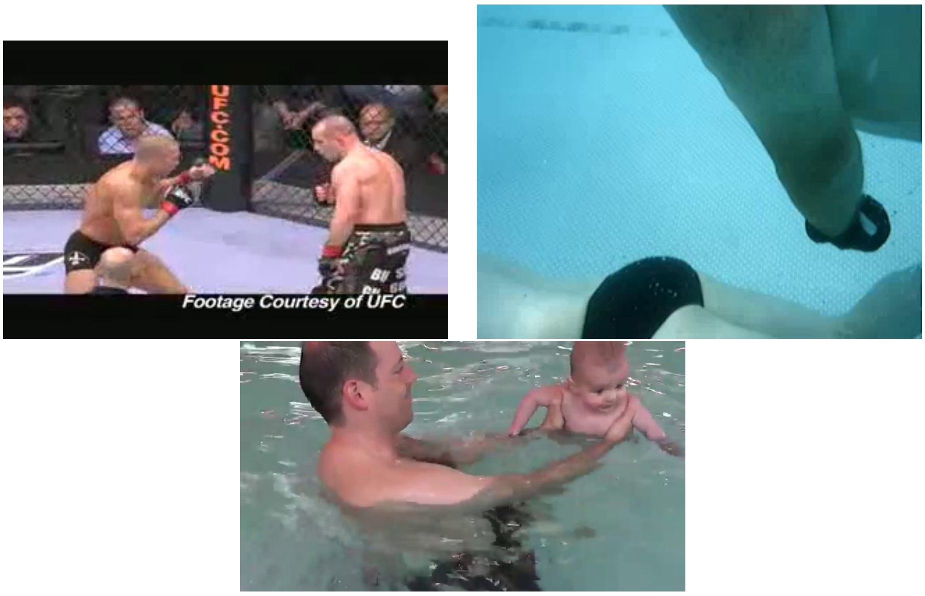


FIGURE 3.4: Wrong predictions from normal boxing, swimming scenes

related problem.

Videos regarding babies (showering or breath feeding.etc) or ones related to sports (swimming, boxing, wrestling.etc) brimming with top-naked scenes are more inclined to be classified as porn.

4. Due to the low resolution, animation videos usually had bad performance, especially Positive cases.

5. Mosaic was added for some pornography videos before formulating the dataset, thus the porn appeared to be less relevant to unhealthy contents for classifier to recognize.

6. Key frame selection performance impeded classification, selected frames were either too trivial comparing to total frame numbers or porn related contents centralized

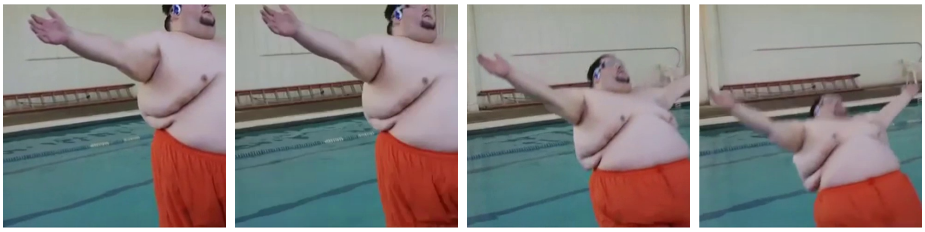


FIGURE 3.5: Wrong predicted frames in naked skin video

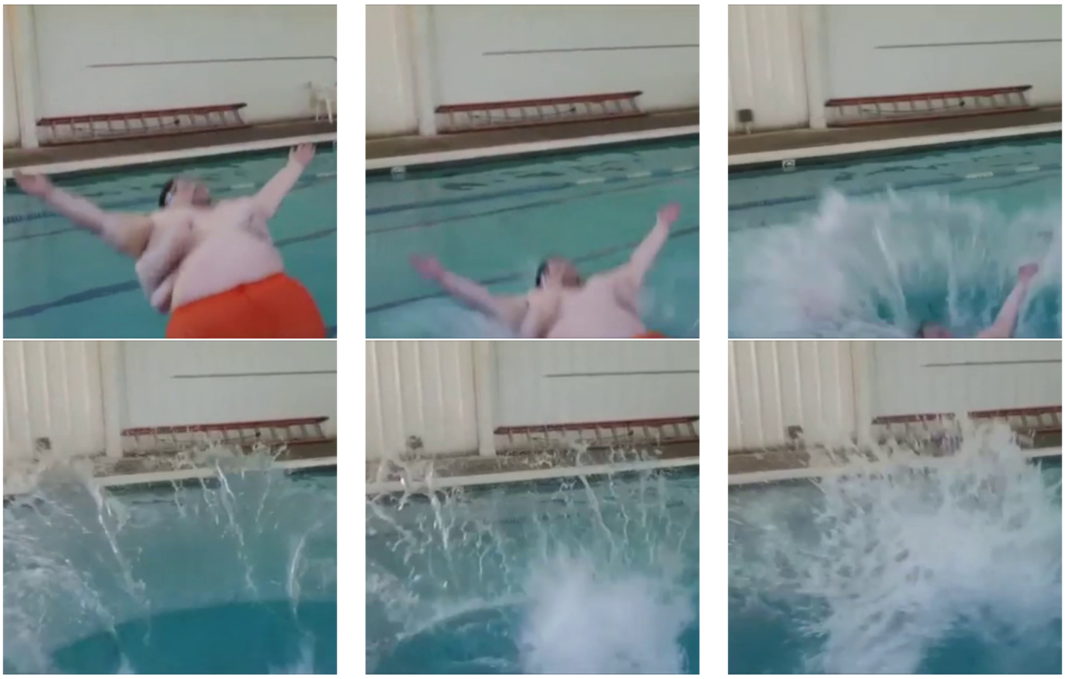


FIGURE 3.6: Correctly predicted frames in a naked skin video

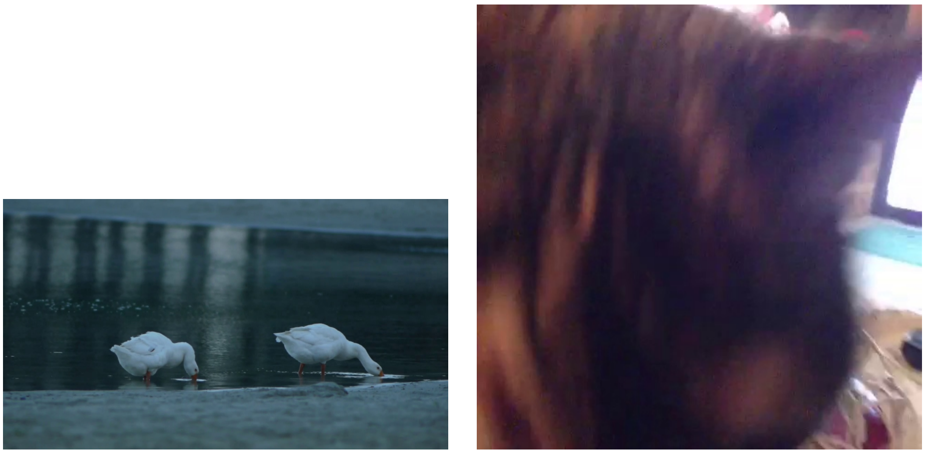


FIGURE 3.7: Examples of frames of animals being predicted as porn

in certain periods that original algorithm tend to ignore. To clarify, some porn videos might have extremely long duration or vast number of frames, although porn scenes were affluent in this video, we only extract 10 frames feeding to image classifier, and these frames happened to not containing porn scenes. In other cases, the selected frames concentrated on only a part of the video, key frame selection was in an nonuniform state.

7. Some porn videos can be regarded as non-pornography, thus the dataset annotation might need modifications.

8. The original video appeared to be dim, the frames selected were extremely indiscernible. This occurred in both positive and negative cases.

3.3 Improvements

Now that we have acquired entire analysis and listed all the potential problems, we could step forward to purpose different solutions to improve the prediction mechanism. Of course, for some of the defects, it would be better to introduce methods aside from computer vision. For example, in some cases, videos start with a scene full of scripts such as ‘warning: this video might contains adult contents’, and because the essence of our Euclidean distance based key frame selection method emphasize intrinsic differences at various time steps, such content with words could be well selected in reality. In this way we could add function of **Optical Character Recognition** (OCR), a widely utilized implementation in optical flow and natural language processing.

On another occasion, as the 7_{th} problem we listed in previous part, each frame was originally dim and blurry continuously in the entire content, so no matter what strategies would be applied to the basis of image classifier, we could not achieve better performance. Instead, audio level information could be taken into account to identify the attribute of these video clips.

In order to enhance predictive performance, strategies were able to be modified from three levels, correspond to image classification, decision making and key frame selection levels.

3.3.1 Image Classification Level Improvements

ia classifier is an image classifier API, according to our previous experiment, we found out that *ia classifier* might be problematic under certain circumstances. For instance, *ia classifier* did not have a decent prediction on animations. Different kinds of situations exist on cartoon videos in **Pornography-2k** dataset. Some are extremely short, cut out from pixel video games with low resolution. Some are fragments of formal cartoon with a duration of nearly ten minutes. A classifier called *CVP* was trained and took features of animations in consideration.

There are similar improvements on *CVP* than *ia classifier*, since pixel level feature from positive videos in **Pornography-2k** dataset was extracted during its instruction. So it was reasonable to conclude that *CVP* might have better performance on predicting pornography videos as positives.

We could feed false predictions generated by *ia classifier* into *CVP*, and observe the possible different predictive pattern of the latter. By distinguishing the advantages and disadvantages of these two classifiers, we were able to combine these predictions and train ensemble models to improve the performance. We kept correctly predicted results from *ia classifier*, extract wrongly corrected videos from *json* work logs, and feed the same frames created by Euclidean key frame selection in these wrongly predicted videos into *CVP*.

When feeding 45 false predicted (28 false positive and 17 false negative) cases of 500-video test set by *ia classifier* in the previous page into the combination of *CVP* and efficient Euclidean, 27 false positives and 1 false negative remained wrongly predicted. We substitute the 45 wrong results with the newly generated ones by *CVP*.

Gathering with performance enhancement on prediction of positive porn videos, *CVP* did had a better performance on animations, whereas the leftover cases which filled with top naked were still problematic and can't be recognized by existing classifiers as normal content. But high True Positive rate indicated that ensemble method was applicable to combine *CVP* with *ia classifier*.

Ensemble

First, acquire results from both training and test set dataset with Euclidean based key frame selection and *ia classifier*. All the detailed results should be restored such as frame indices select in each video, decimal prediction of each frame, binary prediction for each video, etc. The method we used in population generation part was destined to be random in indices. Thus, different experiments yielded different combinations of key

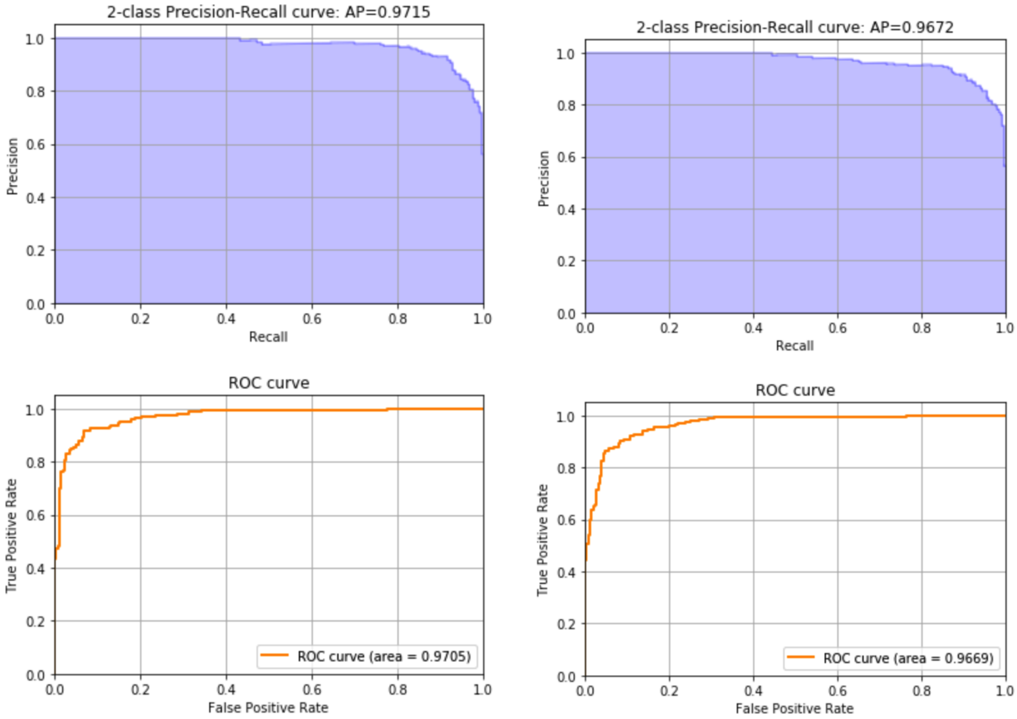


FIGURE 3.8: Average Precision and AUC for ensembles

frame selections. In order to develop a meaningful ensemble method, we must make sure the prediction given by *CVP* for each frame was based on same key frame indices. After defining an object of image classification, we iterated through out these three datasets and pinpointed to corresponding frame indices, read the related frames and feed into *CVP* to reproduce the results.

For one method, we generated features for training set based on average score of *ia classifier* and *CVP*.

For same video, we acquired 2 lists of decimal predictions from the two classifiers, corresponded to the same indices with respectively. Next, for each frame, we calculated average to represent the final decimal prediction.

Now that we had a brand new prediction for each frame of each video in training set (1000 videos with half porn and half non-porn), we could generate features for training a new ensemble model. For each video, the max, min, median as well as mean score was extracted from the brand new scores, forming feature of dimension of 4×100.

In another way of feature generation, we concatenated results from the two classifiers, so for each video, the frame selected was doubled. Next, we collected max, min, median, mean following the same process as the first generation method, to create a 4×100 feature in an expansion.

According to our feature formation, some estimators were applicable for this experiment. Linear SVM as well as Decision Tree fitted training set perfectly, but performance on test set showed a pattern of overfitting.

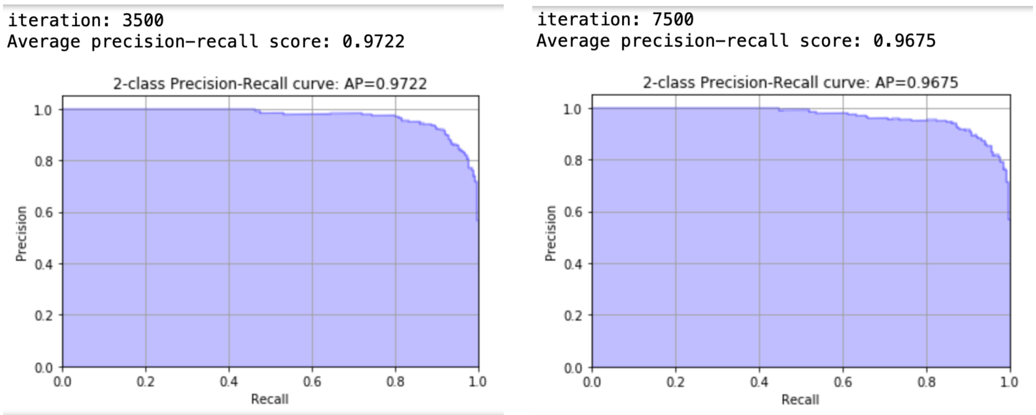
SGD Linear classifier fitted perfectly for arrays of floating point values features. The maximum number of passing over the training data (epochs) was set to 10000.

If no improvement is made during iteration, the default number of early stop was set by default as 5. We used this estimator to fit training data.

The results for these two ensembles on test set:

Method	AP	AUC	Accuracy	Recall	Precision	F1
SGDClassifier-average	0.9715	0.9705	0.92	0.908	0.9302	0.919
SGDClassifier-concat	0.9672	0.9669	0.9	0.936	0.8731	0.9035

We could also alter iteration number to find optimal. For average ensemble, Average Precision could reach maximum of 0.9722 at its 3500_{th} iteration, whereas concatenation ensemble could reach maximum of 0.9675 at its 7500_{th} iteration.



```
iteration: 3500
Average precision-recall score: 0.9722
```

```
iteration: 7500
Average precision-recall score: 0.9675
```

FIGURE 3.9: Best Average Precision for the ensembles

3.3.2 Decision Making Level Improvements

Decision making level improvements mainly aim to address the noise problem existed on analysis section. This established based on our default metric of using maximum as video classification representation.

This part was designed to eliminate False Positives due to noisy frames as much as possible. However, some pornography might be originally correctly predicted because only one of ten frames was given a decimal number of over 0.5. Thus, related experiments might sacrifice True Positives. This is a classification level modification so it would be tricky because we need to look back through image classification level improvements. Luckily, noises in positives occur far more less than noises in negatives. This is the foundation of validity of decision making level improvements.

Noise Elimination

The most direct way to improve on decision making level is adding a conditional statement. If noise exists, use average score to represent the result. We first implement a noise screening. Since for our max metric, a threshold 0.5 was set, if the set of prediction for a single video only contains a single frame score higher than 0.5, we would consider this case as noise. We delete the max frame score, regardless of whether the video was a positive or negative. In other words, the attribute of noise cases is, one of ten selected frames had large score (above 0.5) comparing to the rest. Next, we calculated the average of remaining decimal predictions, and use that value to represent video decimal prediction. Finally, according to whether its value was above or below the threshold, we updated the video binary prediction.

Neighboring Examination

Another situation should be taken into consideration. Due to the essence of Euclidean selection, similar scenes might be excluded. In this way, if only a single frame from an extremely short but non-trivial and non-neglectable period was chosen, it might indicate that pornography exists even though a noisy frame was identified.

Under this circumstance, we considered implementation of a noise neighboring examination. It observed the neighbourhood of the max frame, if this neighbourhood has an average score above 0.5, we tend to predict this video as porn. Specifically, we pinpointed to the index of the noisy frame, set it to be the center, and read the 5_{th} and 10_{th} frame prior and posterior to the noise. We feed these four additional neighbours into image classifier. We calculated average score of the center and two five-unit distance frames, denoted as S_{n1}, as well as average score of all the five frame, S_{n2}. If both numbers are above the threshold, we will update the decimal prediction of this video with the average of S_{n1} and S_{n2}. Of course the binary prediction will be updated to 1.

If S_{n1} is above the threshold whereas S_{n2} was not, we reevaluate the 2_{nd} frame prior and posterior to the original center, repeating the classification process. To clarify, let us define area from the 5_{th} prior to 5_{th} posterior as original neighbouring area. This time, for convenience and approximation, we neglected areas outside original neighbouring area. The average of the 2_{nd} prior, the 2_{nd} posterior, the 5_{th} prior, the 5_{th} posterior was denoted as S_{n3}. We directly looked into S_{n3}, if larger than the threshold, it will be updated as porn.

Method	AP	AUC	accuracy	recall	precision	F1
Baseline	0.9683	0.9634	0.91	0.932	0.8927	0.9119
Noise elimination	0.9557	0.9444	0.908	0.864	0.9474	0.9038
Neighbouring	0.9680	0.9634	0.914	0.908	0.919	0.9135

No matter noise elimination or neighboring examination was implemented, an overall enhancement could not be achieved. As mentioned in 3.6.5, we already concluded a preference for a higher recall. From this perspective, neighboring examination showed a better performance than noise elimination. Noise elimination happened based on the result of baseline, but for neighboring examination, more calculating resources were required for reading new frames and making new predictions. These two methods were not valid for improvements.

From AP and AUC we recognized an overall on performance, this was because we ignored whether the noises came from positives or negatives in the first place. Thus, more True Positive cases were obtained by noise than we originally thought of, that is, a series of porn only had one frame predicted as porn selected, thus ascribed as positives. However, only considering the noises in non-porn is invalid and meaningless, this added to the inoperability of these methods in reality.

Noise Examination on Result Extension

A more complicated method is to extend the selected frames and observe if there exists a better representation of video score based on the prolonged results. In this case we still neglected whether the noises belong to positive or negative by Euclidean selection and *ia classifier*.

For videos identified as abnormal after noise screening, we run the whole Euclidean selection and *ia classifier* prediction, to collect another 10 frame scores so that we can extend the results of noises by twice. It was necessary to pointed out, because of the randomness of selection, one frame could be selected for multiple times, so the extension was not strictly twice.

For noises, we examine the extended scores and assign new values p to them. For a specific noisy video, we extract the scores above 0.5 from the extended frame score, and the new value p is the average of these above 0.5 scores.

The optimal threshold in our case was 0.7, it provided a best performance on both test

```
Noise examination on result extension
Dataset $D$;
Video with index $V_i$;
Result list (List of lists, with each element being a list of 10 frame scores for each video from previous result) $S$;
List of previous predicted scores for each video in the set $PList$;
List of previous binary predictions for each video in the set $PListBinary$;
A Video Classification object $V_c$.
An updated $PList$
for $i\ in\ range(len(D))$ do
  if $len([item\ for\ item\ in\ S[i]\ if\ item\ >\ 0.5])\ ==\ 1$ then
    $res \leftarrow \mathrm{V}_c.predict(\mathrm{V}_i)$
    $S \leftarrow S.extend(res)$
    $p \leftarrow [item\ for\ item\ in\ S\ if\ item\ >=\ 0.5].mean()$
    $PList[i] \leftarrow p$
    if $P < 0.7$ then
      $PListBinary[i]\ =\ 0$
    else
      $PListBinary[i]\ =\ 1$
```

set and validation set, we update the binary predictions for these noises, above which is ascribed to porn whereas below ascribed as non-porn. Among 33 noises in the test set, 17 were porn, 16 were non-porn. After noise examination and new binary prediction with threshold being 0.7, 4 false negative and 4 false positive cases were generated. Among 30 noises in the validate set, 12 were porn, 18 were non-porn. After noise examination and new binary prediction with threshold being 0.7, 4 false negative and 3 false positive cases were generated.

Method	accuracy	recall	precision	F1
test set with noise examination	0.926	0.916	0.9347	0.9253
val set with noise examination	0.914	0.9	0.9259	0.9128
test set without noise examination	0.91	0.932	0.8927	0.9119
val set without noise examination	0.892	0.9165	0.8740	0.8945

According to the result above, overfitting was not obvious, and overall performance could be enhanced by noise examination on result extension.

3.3.3 Frame Selection Level Improvements

This was in accordance with problem 6 in analysis section, and we mainly wanted to decrease False Negative rate. In other words, we wanted to identify more porn videos. Note that since we used maximum as metric, and a threshold was given to ascribe the video into porn or non-porn. False Negative meant all the key frame selected were given scores under the threshold.

In genetic algorithm, the feature we used during calculation was Euclidean distance between frames. We examined color distribution of the entire video stream, particularly we calculates color value for each frame selected. The color function indicated the percentage of naked skin in the selected frame. In return, the indices group we chose to feed to image classifiers not only guaranteed the largest distance, but also maximized the possibility of choosing porn scenes from positive cases.

However, unlike decision making level improvement, where noisy True Positives might have minor impact on the decrease of False Positive rate. Under this circumstance, there were even more amount of non-pornography videos with naked skin contents like we mentioned from problem 3 in the analysis part than False Negatives with problem 6. In detail, in the test set of 500 videos, there were 17 False Negatives. However, out of 28 False Positives, 22 had problem 3. The fact was that problem 3 and 6 were conflicted with each other. Thus, in theory, skin color method could bring about more False Positives. When implementing genetic algorithm, we only took the summation of Euclidean distance into account. For skin detection, we added a color function with normalization and calculate color representation to each step, to finally select the combination of 10 frames with largest distance as well as skin exposed area. Color function majorly calculated the percentage of skin area in a selected frame.

Recall that we converted frames to 256 × 256 when calculating Euclidean distance and utilized greyscale. This time converted frames to HSV color space. H, S, V represents hue, saturation and value with respective. Hue expresses basic properties, portion of colors in different dimensions. In **opencv**, it is displayed within 0 to 180. Saturation

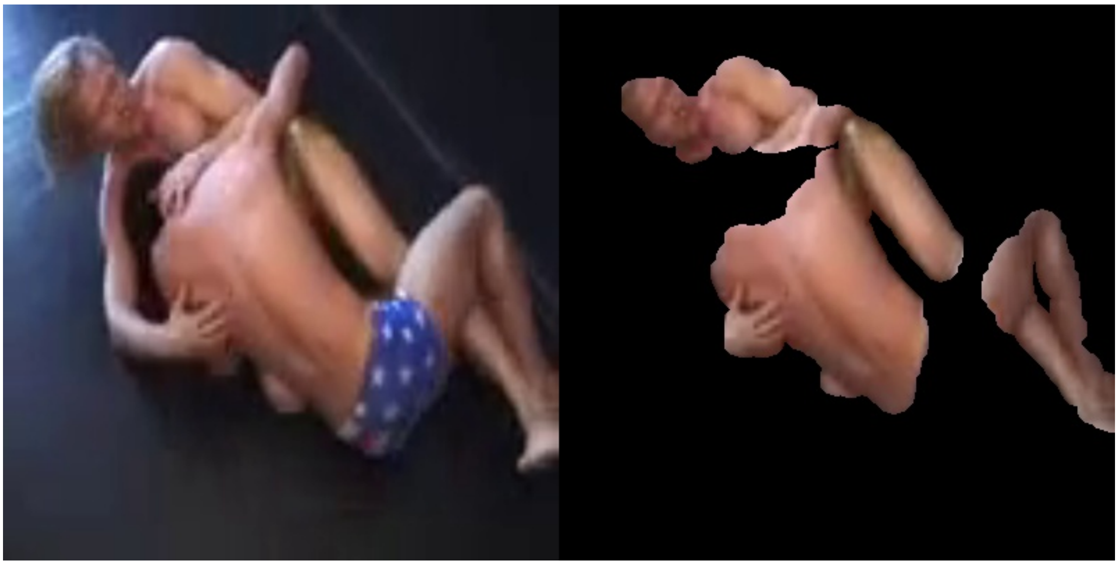


FIGURE 3.10: Skin area detection for a wrestling scene

represents greyness in color space. Value displays brightness and intensity and interrelates with saturation.

In HSV space, human skin can be approximated to upper and lower limits as pixel intensities. In detail, it sets minimum and maximum value for all the three spaces H, S and V. We created a binary mask, a single channel image, appointing satisfactory pixels fall into the upper and lower bound of skin. Skin detection occurred essentially in the conversion to HSV and mask creating steps. After masking, pixels whose values were 255 (white) indicated skin pixels in reality, whereas 0 (black) being non-skin area. To remove tiny regions which might be false positive skin, we applied erosion and dilation with a 10×10 square kernel. Gaussian Blur was also implemented for the purpose of smoothing, allowing our mask with more sanity.

Finally, we counted non-zero pixels, and divided by the frame area, which was 256×256, to acquire the final percentage representation of skin area in each frame.

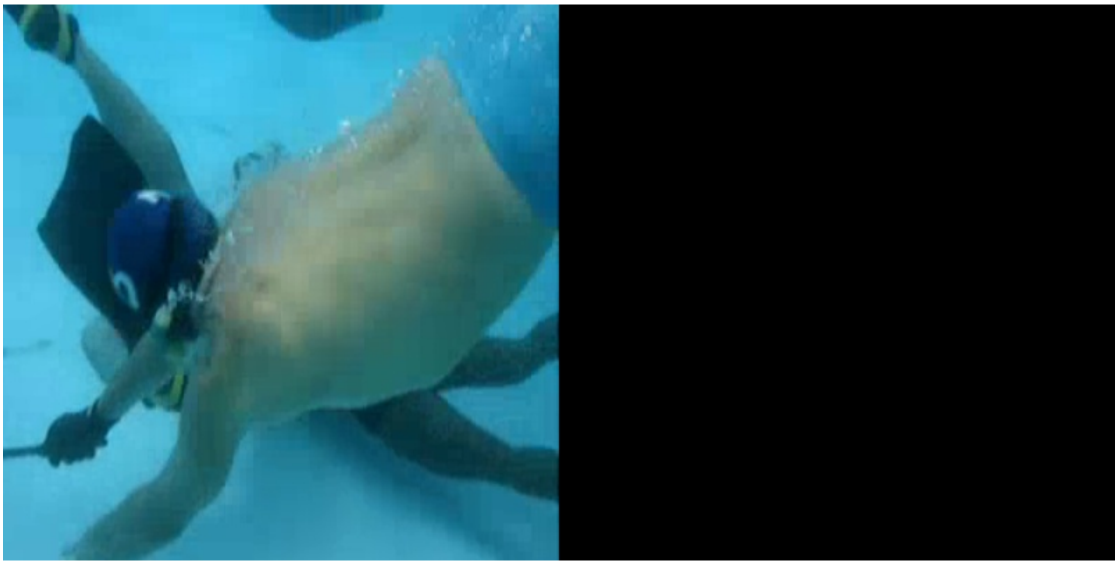


FIGURE 3.11: Limitation of skin area detection

However, such skin detection method was not without its limitations. There are some pretty obvious limitations and drawbacks to this approach. We ascribe this skin related problem to color detection. But the pixel intensities range of a skin was chosen subjectively, an optimal threshold did not exist.

Also, with disparate frame resolution and lighting flows, the performance might vary. In our experiment, we set the lower bound and upper bound of H, S, V dimension as [0, 40, 90] and [20, 255, 255], providing decent results for most videos. It is important to point out though, for specific cases, skin area might not be identified at all.

After adding skin area into genetic algorithm, we ran the whole frame selection and image classification together (with *ia classifier*), the result validated the conflict of problem 3 and 6 that we stated previously. Truly, this alleviated the misclassification of positive videos, thus False Negatives decreased, but since problem 3 existed more widely in the dataset, False Positives increase on a larger basis. In this way, Accuracy would be sacrificed inevitably. Recall and Precision would act reversely comparing to **Baseline**. As usual, max was the representation of video score.

Method	AP	AUC	Accuracy	Recall	Precision	F1
Baseline	0.9683	0.9634	0.91	0.932	0.8927	0.9119
Skin Detection	0.9437	0.9459	0.89	0.952	0.8470	0.9463

3.4 Conclusion

Euclidean distance based key frame selection did have improvement on the result comparing to random selection. Due to its difference dominated idea, scenes of various contents would be extracted so more factors feeding to image classifiers would be considered. Among all the improvements, ensemble provided best performance enhancement. Noise examination on result extension was also operable, but key frame level improvement was hindered because of the nature of naked skin existed in negative cases.

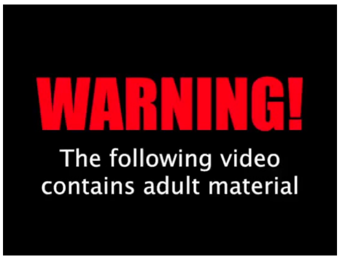
Chapter 4

Future Works

4.1 Pre-classification and NLP Level Improvement

As mentioned in previous analysis, some of the problems were not able to be solved by image classification. For instance, due to the darkness and vague of the video files itself, classifiers were not able to find out critical contents and decisive features that allowed it to make reasonable prediction.

However, thanks to key frame selection, we were able to collect some informative features for videos with bad quality. Some of them directly indicated the property of the video, as shown in FIGURE 4.1. For some cases on the other hand, the information was not as obvious or decisive.



WARNING!

The following video
contains adult material

FIGURE 4.1: Some frames selected contain direct message

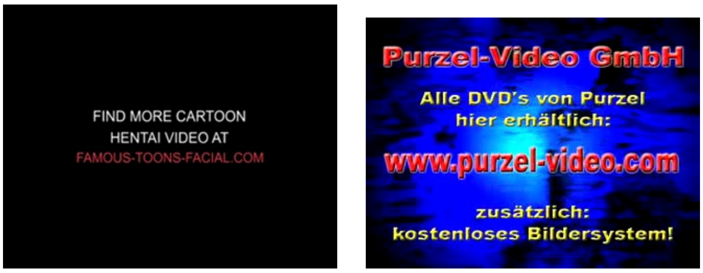


Figure 4.2: Other cases when language information was involved

Thus, when we take such situations into account, we would like to inspect if a frame selected belongs to word expression from scratch. If we could extract the gist of such a scene, we might do some light weighted semantic analysis, which belongs to Natural Language Processing level improvement.

Continuing our frame selection logic, we can still obtain 10 key frames for the video. Next, we could implement another simple image classifier, for the images predicted as word related, we could directly proceed with semantic analysis. If the image contains information similar to the one in FIGURE 4.1, we can skip feeding other remaining frames to pornography binary image classifier and obtain a prediction in a more direct way.

4.1.1 Transfer From PDF Element Classification

From another project, we had an object of extracting effective information on paragraphs in Portable Document Format (PDF) files. The preliminary steps were to identify different types of contents, and to proceed different operations for different parts.

Starting the task, we collected a series of PDF to be analyzed, formed a 400-page PDF

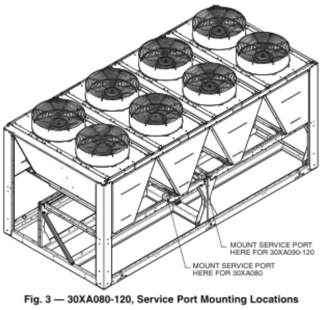
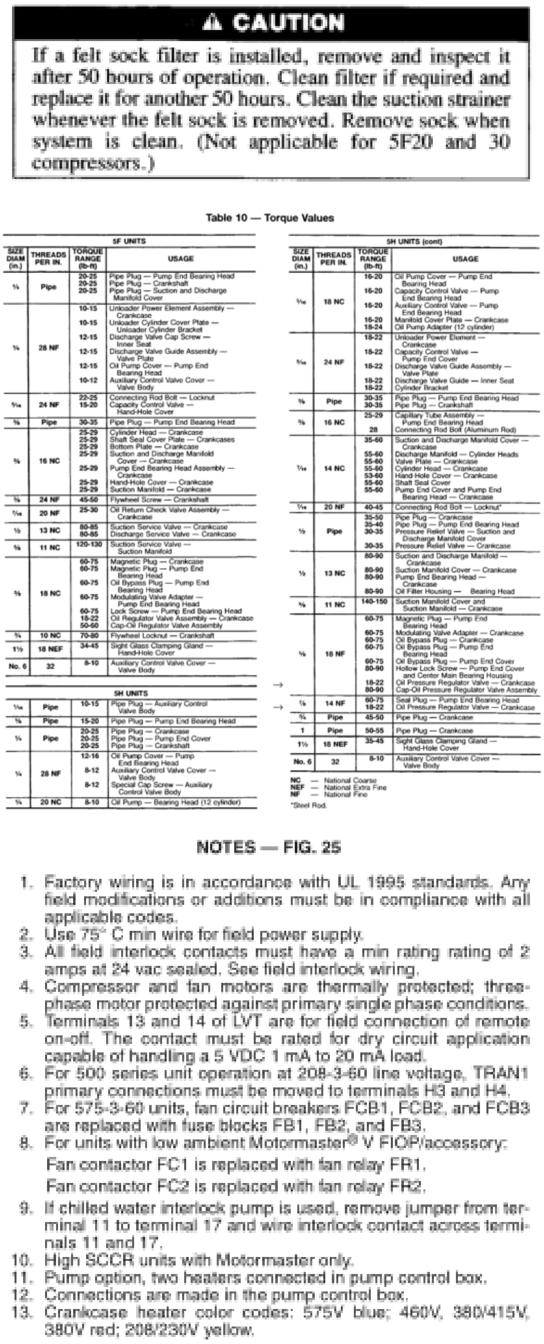


FIGURE 4.3: Figure predicted by the PDF classifier

data collection, which were irregularly formatted with different types of contents. Each page from all the PDF was first converted to *jpg* as well as greyscale for further image cropping. The images used for training was divided into texts, figures, covers and tables as 1:1:1:0.85, respectively.

Second, we implemented a connected graph based method to cut each PDF into different segments. When appropriate dilation is implemented, we could pinpoint to different coordinates to obtain the tiny image crop for training a classifier. We manually annotated the cropped segments into four categories, texts, figures, covers and tables. Utilizing the created crops, we constructed a Res-Net network in *Pytorch* to train a PDF contents classifier. Cross Entropy was used as loss function. Validation loss was flipping around 0.008 in the end, accuracy from training, validation and testing set were all above 94%. We could adjust this PDF classifier by combining non-figure category into a single class called "words-related", to reform this classifier into a binary classifier. After acquiring key frame results, we implement binary predictions on words. For "words-related" frames, we would proceed with OCR and Semantic Analysis.



⚠ CAUTION

If a felt sock filter is installed, remove and inspect it after 50 hours of operation. Clean filter if required and replace it for another 50 hours. Clean the suction strainer whenever the felt sock is removed. Remove sock when system is clean. (Not applicable for 5F20 and 30 compressors.)

Table 10 — Torque Values

5F UNITS			
SIZE DIAM (in.)	THREADS PER IN.	TORQUE RANGE (lb-ft)	USAGE
⅛	Pipe	20-25 20-25 20-25	Pipe Plug — Pump End Bearing Head Pipe Plug — Crankshaft Pipe Plug — Suction and Discharge Manifold Cover
¼	28 NF	10-15 10-15 12-15 12-15 12-15 10-12	Unloader Power Element Assembly — Crankcase Unloader Cylinder Cover Plate — Unloader Cylinder Bracket Discharge Valve Cap Screw — Inner Seat Discharge Valve Guide Assembly — Valve Plate Oil Pump Cover — Pump End Bearing Head Auxiliary Control Valve Cover — Valve Body
5/16	24 NF	22-25 15-20	Connecting Rod Bolt — Locknut Capacity Control Valve — Hand-Hole Cover
⅜	Pipe	30-35	Pipe Plug — Pump End Bearing Head
⅜	16 NC	25-29 25-29 25-29 25-29 25-29 25-29 25-29	Cylinder Head — Crankcase Shaft Seal Cover Plate — Crankcases Bottom Plate — Crankcase Suction and Discharge Manifold Cover — Crankcase Pump End Bearing Head Assembly — Crankcase Hand-Hole Cover — Crankcase Suction Manifold — Crankcase
⅜	24 NF	45-50	Flywheel Screw — Crankshaft
7/16	20 NF	25-30	Oil Return Check Valve Assembly — Crankcase
½	13 NC	80-85 80-85	Suction Service Valve — Crankcase Discharge Service Valve — Crankcase
⅝	11 NC	120-130	Suction Service Valve — Suction Manifold
⅝	18 NC	60-75 60-75 60-75 60-75 60-75 18-22 50-60	Magnetic Plug — Crankcase Magnetic Plug — Pump End Bearing Head Oil Bypass Plug — Pump End Bearing Head Modulating Valve Adapter — Pump End Bearing Head Lock Screw — Pump End Bearing Head Oil Regulator Valve Assembly — Crankcase Cap-Oil Regulator Valve Assembly
¾	10 NC	70-80	Flywheel Locknut — Crankshaft
1½	18 NEF	34-45	Sight Glass Clamping Gland — Hand-Hole Cover
No. 6	32	8-10	Auxiliary Control Valve Cover — Valve Body

5H UNITS			
SIZE DIAM (in.)	THREADS PER IN.	TORQUE RANGE (lb-ft)	USAGE
1/16	Pipe	10-15	Pipe Plug — Auxiliary Control Valve Body
⅛	Pipe	15-20	Pipe Plug — Pump End Bearing Head
¼	Pipe	20-25 20-25 20-25	Pipe Plug — Crankcase Pipe Plug — Pump End Cover Pipe Plug — Crankshaft
¼	28 NF	12-16 8-12 8-12	Oil Pump Cover — Pump End Bearing Head Auxiliary Control Valve Cover — Valve Body Special Cap Screw — Auxiliary Control Valve Body
¼	20 NC	8-10	Oil Pump — Bearing Head (12 cylinder)
5/16	18 NC	16-20 16-20 16-20 16-20 18-24	Oil Pump Cover — Pump End Bearing Head Capacity Control Valve — Pump End Bearing Head Auxiliary Control Valve — Pump End Bearing Head Manifold Cover Plate — Crankcase Oil Pump Adapter (12 cylinder)
5/16	24 NF	18-22 18-22 18-22 18-22 18-22	Unloader Power Element — Crankcase Capacity Control Valve — Pump End Cover Discharge Valve Guide Assembly — Valve Plate Discharge Valve Guide — Inner Seat Cylinder Bracket
⅜	Pipe	30-35 30-35	Pipe Plug — Pump End Bearing Head Pipe Plug — Crankshaft
⅜	16 NC	25-29 28	Capillary Tube Assembly — Pump End Bearing Head Connecting Rod Bolt (Aluminum Rod)
7/16	14 NC	35-60 55-60 55-60 55-60 53-60 55-60 55-60	Suction and Discharge Manifold Cover — Crankcase Discharge Manifold — Cylinder Heads Valve Plate — Crankcase Cylinder Head — Crankcase Hand-Hole Cover — Crankcase Shaft Seal Cover Pump End Cover and Pump End Bearing Head — Crankcase
7/16	20 NF	40-45	Connecting Rod Bolt — Locknut*
½	Pipe	35-50 35-40 30-35 30-35	Pipe Plug — Crankcase Pipe Plug — Pump End Bearing Head Pressure Relief Valve — Suction and Discharge Manifold Cover Pressure Relief Valve — Crankcase
½	13 NC	80-90 80-90 80-90 80-90	Suction and Discharge Manifold — Crankcase Suction Manifold Cover — Crankcase Pump End Bearing Head — Crankcase Oil Filter Housing — Bearing Head
⅝	11 NC	140-150	Suction Manifold Cover and Suction Manifold — Crankcase
⅝	18 NF	60-75 60-75 60-75 60-75 60-75 80-90 18-22 80-90	Magnetic Plug — Pump End Bearing Head Modulating Valve Adapter — Crankcase Oil Bypass Plug — Crankcase Oil Bypass Plug — Pump End Bearing Head Oil Bypass Plug — Pump End Cover Hollow Lock Screw — Pump End Cover and Center Main Bearing Housing Oil Pressure Regulator Valve — Crankcase Cap-Oil Pressure Regulator Valve Assembly
⅞	14 NF	60-75 18-22	Seal Plug — Pump End Bearing Head Oil Pressure Regulator Valve — Crankcase
¾	Pipe	45-50	Pipe Plug — Crankcase
1	Pipe	50-55	Pipe Plug — Crankcase
1½	18 NEF	35-45	Sight Glass Clamping Gland — Hand-Hole Cover
No. 6	32	8-10	Auxiliary Control Valve Cover — Valve Body

NC — National Coarse
NEF — National Extra Fine
NF — National Fine

*Steel Rod.

NOTES — FIG. 25

1. Factory wiring is in accordance with UL 1995 standards. Any field modifications or additions must be in compliance with all applicable codes.
2. Use 75° C min wire for field power supply.
3. All field interlock contacts must have a min rating rating of 2 amps at 24 vac sealed. See field interlock wiring.
4. Compressor and fan motors are thermally protected; three-phase motor protected against primary single phase conditions.
5. Terminals 13 and 14 of LVT are for field connection of remote on-off. The contact must be rated for dry circuit application capable of handling a 5 VDC 1 mA to 20 mA load.
6. For 500 series unit operation at 208-3-60 line voltage, TRAN1 primary connections must be moved to terminals H3 and H4.
7. For 575-3-60 units, fan circuit breakers FCB1, FCB2, and FCB3 are replaced with fuse blocks FB1, FB2, and FB3.
8. For units with low ambient Motormaster® V FIOP/accessory:
   Fan contactor FC1 is replaced with fan relay FR1.
   Fan contactor FC2 is replaced with fan relay FR2.
9. If chilled water interlock pump is used, remove jumper from terminal 11 to terminal 17 and wire interlock contact across terminals 11 and 17.
10. High SCCR units with Motormaster only.
11. Pump option, two heaters connected in pump control box.
12. Connections are made in the pump control box.
13. Crankcase heater color codes: 575V blue; 460V, 380/415V, 380V red; 208/230V yellow.

FIGURE 4.4: "words-related" clips predicted by the PDF classifier

4.1.2 OCR

Previous text recognition [2] tended to implement traditional algorithms such as **Connected Components Analysis** and **Sliding Window** because of their popularities. Later, variations which focus on smaller segments methods such as **label embedding** and **strokes character Key points** were proposed.

As a significant research realm in computer vision, text detection in images has been ineluctably effected by the thrive of learning based methods. In "post" deep learning era, text recognition possessed a huge advantage comparing to the traditional implementation. Hand-crafted features can be omitted to a large extent, saving us time for scheduling and monitoring.

There exists four types of mainstreams: detection that pinpoint language in natural image, end-to-end method automatically addressing both text recognition and word conversion, identification system that transfer text in region of interests philologic representation, ministrant approaches that assistant to facilitate main job of detection and identification.

Also, **Optical Character Recognition** is actually very robust and ideal, we could also consider calling API to finish this task.

4.1.3 Semantic Analysis

We would like to get the gist of a single frame, semantic analysis might not be referring to the actual terminology notation here, it is a broadly speaking semantic meaning in the sentence we extract. From my point of view, a more accurate expression should be text classification. We want to examine whether a video contains adult contents or not, so the word expressions appear on frames selected should be assigned with a binary text classification task.

We can finetune Google's BERT model on our binary text classification. The difficulty for our pornography lies in the fact that related dataset was not as accessible as yelp ratings or IMDB movie reviews, so we might need to collect erotic languages as well as normal ones for finetuning a BERT model exclusive to pornography.

According to the sentences we collected, we need to transfer the data into *tsv*, a BERT-friendly format, and split them into *train* and *dev* set. Next, we need to convert *tsv* formatted dataset into InputExample objects. And InputExample also needs to be converted to InputFeatures object by a function called convert_example_to_feature. After preprocessing, we will be able to tune the model. Thus, obtaining our binary prediction on the word expression frame.

4.2 Multi-Feature Fusion

The application of previous part provides a decent solution to word explanations that exist on the video raw data, this would be useful for addressing the problems 4 and 8 listed in analysis part. There are many cases of animations in small script video game, and can somewhat release the bad prediction rate for non ideal low resolution videos. But sound information is inevitably more profoundly available in pornography. We can take advantage of this feature of raw data onto further enhancing recognition rate.

To achieve this, we need to extract audio features from raw video.

Multiple features could be extracted by traditional communicational techniques [3]. We first convert videos into accessible sound files, and resample the audios to **Short Time Fourier Transform** spectrograms. Next, harmonic ingredient should be isolated from percussive utilizing Harmonic Percussive Source Separation [4]. We need to remove the lower frequency component by transforming information on **STFT** to **Mel**, and finally obtain **Mel Frequency Cepstral Coefficient** spectrum from **Log-Mel**. Finally, when such features are available, we generate features resemble images which are able to be fed into deep learning architectures. we apply convolutional neural networks. **Log-Mel** level information would be able to feed to CNN. We could specify 2 classes, also diverse as porn and non-porn. The fusion process is completed via choosing the model with best performance for both of these classes.

Feature Fusion could be also combined with frame selection, we could exploit the frame indices generated for each video, and to focus on nearby audio features. That is, we could implement sampling and resampling techniques considering our frame selection mechanism. Based on the combination of pixel level frame predictions as well as audio level frame predictions, we could implement ensemble to further improving the performance of our model.

4.3 Video Level Training Approaches

We could enhance the pornography recognition rate from a more macroscopic angle. It is no doubt that image classifiers are easier to train, comparing to directly scrutinize video files. With the help of the state-of-art architectures in deep learning, we might implement video level analysis directly when computational condition is satisfied. For current stage, this might not be accessible due to the distribution of computational resources.

4.3.1 Deep Feature Flow

We might consider to introduce deep feature based CNN. Convolutional neutral networks thrive during this decade and showed a breakthrough in computer vision task, especially image recognition. However, there remains a bottleneck for implementing the state-of-art network of video streams. Because of the affluence of high resolution video resources, operating a video by frames requires a huge demand on computational resources and processing time. Complexity of processing videos could require exponential resources comparing to feeding the pixel level image information to deep convolutional neural networks.

Deep feature flow based networks were put forward [5], allowing us to only keep track of the sparse information inside the key frames. It provided guarantee in terms of speed as well as flexibility for video recognition. Flow computation is a traditional approach quicker than convolutional characteristics calculation. We can make estimation when taking flow field characteristics into account, allowing us to develop a user friendly end-to-end network, optimizing video identification from image evaluation and flow field perspectives. It provides us with the capability of processing flow training and video level examination simultaneously.

Optical flow, a traditional concept in modern information retrieval tasks, is the foundation of video evaluation. Most previous papers focused on variations and researchers took much energy on displacement. When deep semantic was combined with the mainstream of optical flow, we can address motion measure, pose estimation with decent performance, which would be very useful for distinguishing erotic postures in pornography.

First, we would make inferences on deep feature flow. Two networks should be built. The first one is responsible to creating feature maps from a fully connected layer, the other takes charge of video recognition of the feature maps generated from the first network.

There is a huge tendency towards repetition of feature maps of neighboring video frames as we expected. Like our baseline, Euclidean based key frame selection, we examine similarity between frames so that the feature maps could be extracted from sparse information matrix instead.

If a following frame contains small similarity with its prior, it would not be considered with priority and its feature map would be transferred from its prior frame. Specifically, the spatial consistency from nearby frames provides us with the possibility of propagating features easily.

We would like to obtain flow field by estimation. To achieve this, we can assign an image to its neighbor frames among the video stream [6], which is a task of image registration. Next, we match SIFT features from the neighboring frames while conserving discontinuity in the video stream.

We can also convert this problem to a supervised learning [7]. Flow estimation requires us to extract consistency between images, so we also need to learn to match different scenes [8]. This is different from traditional usage of convolutional neural networks in computer vision tasks. We could implement a layer which allows matching to calculate the correlation between adjacent frames, so that it could learn how to estimate flow from matching result.

4.3.2 Adaptive Scan

We mentioned multiple times from previous content that adjacent frames contain more similarities. Thinking this from another perspective, we could say that only a little proportion of frame could contain all the information we need for binary video classification. One of the downsides of AdaScan is, for now we could not find a pretrained model pretty applicable for our problem, so training from scratch is unavoidable. We could adaptively and continuously provide predictions on discernible significance of each frame, and select them in sequence.

This approach [9] stands out from following perspectives: It only scrutinizes the most informative frames, in terms of human body actions which aligns our pornography recognition. It is end-to-end trainable with respective to image labels, so the final goal of binary classification is operable. It is an inductive approach so that the parameterization of information pooling is retrieved automatically, and the entire training set was not a necessity from perspective of testing. It outperformed related methodologies in human motion evaluation, and could be implemented with complimentary symbolization of video streams.

AdaScan is special because of its pooling structure, it looks through the entire video, actively pools features of all the frames inside and provide a vector as summarization. The generalization of the vector accommodates to action recognition, in our task, to be pornography recognition.

It is made up of three connected logical parts, completing feature extraction, adaptive selection and binary prediction with respectively.

The feature extractor simply collects features of each frame and forms normalized vectors correspondingly. Pooling structure recursively makes predictions on how much a frame is required to be the most essential content on our motion analysis task.

Two factors were taken into account to decide significance of frame from current time step. The first is the feature extracted by the previous part of AdaScan, the other is the vector generated up to date.

A mechanism is introduced to give evaluation on then updated vector. The discernible feedback is assembled merely via pooling the frames that have been already chosen. Final vector is generated adaptively following these steps.

We can implement normalization on the final result for robustness, and the binary pornography prediction can be acquired after utilizing a softmax Fully Connected layer at the end of the structure.

Solutions to this part have a high requirement on computational resources, the best strategy for short are to focus on image classification level improvements and to optimize on existed model.

Bibliography

[1] Xue Yang, Zhicheng Wei, Genetic Keyframe Extraction for Soccer Video, PEEA, 2011: 713-717.

[2] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. arXiv preprint arXiv:1811.04256, 2018. 1, 2.

[3] Jisheng Bai, Chen Chen, et al. Urban Sound Tagging With Multi-Feature Fusion System, Detection and Classification of Acoustic Scenes and Events, 2019.

[4] Sumair Aziz, Muhammad Awais, Tallha Akram, et al. Automatic Scene Recognition through Acoustic Classification for Behavioral Robotics, Electronics 2019, 8, 483.

[5] Xizhou Zhu, Yuwen Xiong, et al. Deep Feature Flow for Video Recognition, CVPR, 2017: 2349-2358.

[6] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: dense correspondence across difference scenes. In ECCV, 2008.

[7] Philipp Fischer, et al. FlowNet: Learning Optical Flow with Convolutional Networks, 2015 IEEE International Conference on Computer Vision (ICCV).

[8] Ce Liu, Jenny Yuen, Antonio Torralba, et al. SIFT Flow: Dense Correspondence

across Different Scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 33, Issue: 5, May 2011.

[9] Karan Sikka, et al. AdaScan: Adaptive Scan Pooling in Deep Convolutional Neural Networks for Human Action Recognition in Videos, CVPR, 2017.