Methods for pattern detection based on colour and size features in video streams

Nguyen Cuong The1,2, V I Syryamkin1 and Nguyen Trang Hoang Thuy1,2

1Institute of Applied Mathematics and Computer Science, Tomsk State University, Tomsk, Russia
2Faculty of Information technology, Vietnam Maritime University, Haiphong, Vietnam

E-mail: cuongntit@vimaru.edu.vn

Abstract. Detecting pattern in images plays a huge role in the field of computer vision. The article summarizes methods for detecting patterns by their properties, such as colours, shapes in images. And with video, the detection of patterns is also based on their movement. Human face recognition problem is also mentioned in the application of MTCNN network architecture and reconfiguring FaceNet to suit actual requirements. Test results are given and recommendations with data and program operating conditions are provided for the reference.

1. Problem of pattern recognition in video stream

1.1. A brief history of pattern recognition in images

The pattern detection in images has been studied since the 1960s with studying the cat's brain response to physical vision [1]. Then in 1963, Larry Robert in [2] simulated the patterns in the image in "blocks world". In 1970, David Marr [3] gave the basic sketch lines for 3D pattern identification. In 1973, Fischler and Elschlager introduced a method of identifying people through special points in various positions of the body [4]. In [5], Brook and Binford (1977) simulated human shapes through basic cylinders. In [6], simple lines were used to connect points surrounding the pattern. Shi and Malik (1997) have locally differentiated image regions to present an image area containing patterns [7]. In 1999, David Lowe introduced the SIFT function to identify deformed pattern in different images [8]. A great achievement in human face recognition was given by Viola and Jones (2001) based on Haar-likes features [9]. In the early 2000s, along with the development of machine learning and deep learning models [10], standard image datasets were created to allow to measure the effectiveness of identification methods.

1.2. Challenges in pattern recognition

The semantic gap: The problem of semantic distance: The computer "understands" about the pattern in the image as just a matrix of pixels with RGB colour channel values. So the matrix of image changes (even if only a pixel), it becomes another "image matrix". Meanwhile, optical perception by human eye is not much various.

Illumination: Although images contain the same pattern, but with lighting conditions going on in the scene. The problems are: dark light, bright light or back light.

Deformation: Patterns can be deformed (typically animals), they can really assume a lot of different, varied poses and positions.
**Occlusion**: Patterns may appear incompletely in images, with only part of the body or be obscured. So pattern recognition image processing algorithms need to be powerful in this case.

**Background clutter**: The foreground of the pattern would actually be able to look quite similar in appearance to the background.

**Intraclass variation**: That is one notion of pattern-ness, actually spans a lot of different visual appearances. And patterns can come in different shapes and sizes and colors and ages. So that algorithm, again, needs to work and handle all these different variations.

### 2. General theoretical issues and pattern detection method

In terms of general perception, the position of the pattern in the image is the image area that is "distinct" from the background. The processing time to detect the pattern in the frame must be less than the delay time the frame is shown (the FPS parameter in the video). The CNN model of stringed neural networks proposed by LeCun et al. in 1989 [11] opened up a new research direction in detecting patterns in images as well as in video (Figure 1) [12].

![Figure 1. Typical CNN architecture.](image)

A convolution neural network consists of an input and an output layer, as well as multiple hidden layers. CNN's hidden layers usually consist of a series of complex layers that can vary with convolution [12]. The trigger function is usually ReLU layers, and then following by additional convolutional parts such as layers of convolution, pooling and fully connected layers. Therefore, they called hidden layers because of their inputs and output are obscured by the active functions and final convolution operators.

#### 2.1. Pattern detection is based on area proposal

This method is based on the principle of dividing an image into different regions, patterns that exist in image areas with saliency properties. In 2014, the model R-CNN [13] was proposed to divide the image into 2000 regions for convolution to detect the pattern. In 2015, model SPP-net [14] was introduced to overcome the limitation of R-CNN by adding SPP layers at the end with flexible sized proposed regions, which increases the computation speed of model. However, the limitation of SPP-net is the large storage space and the transition layers before the SPP layer cannot be tweaked. Fast R-CNN model [15] and later Faster R-CNN [16] in 2015, with the adjustment of the multitasking loss function in classification and limited regression, combined with region of interest (RoI) with predefined anchor boxes in sliding window splits to produce fixed-length vectors taken by convolution and regression layers to obtain the corresponding output. Improvements in the RoI layer were given by models R-FCN [17], FPN [18] and Mask R-CNN [19]. The following Table 1 [20] compares the performance of algorithms based on average precision (mAP), test time, and frames per second (FPS) rate. Data were obtained from the PASCAL VOC 2007 - 2012 test set [21].

**Table 1. Comparison of test general on test datasets VOC07 and VOC12.**

| No | Algorithm | Train on | mAP (%) | Test time (sec/ img) | Rate (FPS) |
|----|-----------|----------|---------|---------------------|------------|
| 1  | R-CNN [13]| 07       | 66.0    | 32.84               | 0.03       |
| No | Algorithm         | Train on | mAP (%) | Test time (sec/ img) | Rate (FPS) |
|----|-------------------|----------|---------|----------------------|------------|
| 2  | SSP-net [14]      | 07       | 63.1    | 2.30                 | 0.44       |
| 3  | FRCN [15]         | 07+12    | 66.9    | 1.72                 | 0.60       |
| 4  | Faster R-CNN [16] | 07+12    | 73.2    | 0.11                 | 9.10       |
| 5  | R-FCN [17]        | 07+12    | 83.6    | 0.17                 | 5.90       |
| 6  | YOLO [22]         | 07+12    | 63.4    | 0.02                 | 45.00      |
| 7  | SSD 300 [23]      | 07+12    | 74.3    | 0.02                 | 46.00      |

The comparison result table shows that the algorithms work well with single images, can recognize patterns of small size. However, with a low FPS, it is not suitable for applying to pattern recognition in video. The algorithms YOLO [22] and SSD [23] will be discussed below.

2.2. Pattern recognition based on classification and regression

The detection method is based on area proposal that performs well with discrete images. Then for each image to be processed, categorization and limited regression are often trained individually. Therefore, the processing time of different components makes the total processing time of an image lengthy, not suitable for real-time applications.

To reduce performance time costs, the network models based on regression/classifications, which map straight from pixels to bounding box coordinates and probability layers. The next content will outline the historical models and focus on two models that are currently quite well evaluated, YOLO and SSD.

For the purpose of determining the proposed limit boxes correctly, the authors have implemented a regression (repeating multiple) method with multiple parameters to localize bias. In 2014, MultiBox model was proposed by Erhan et al. in [24] to predict classification limit box coordinates. However too many parameters are passed into the last layer. In 2015, Yoo et al. [25] proposed the AttentionNet model with scanning the entire image, creating the weak directions quantized and converging to a precise pattern boundary box with a predictable set repeat. However, that model proved ineffective when handling many categories. In 2016, Najibi et al. [26] proposed a method of pattern detection based on fixed multi-scale bounding box grid repeated to scan the entire image and scale the elements of the mesh, calling is G-CNN. However, it is not very useful for dealing with overlapping or small-sized patterns.

In 2016, Redmon et al. [22] launched the YOLO model. At first, the image is divided into an \( S \times S \) grids. If the grid center is in the grid cells, the grid will be able to detect the pattern. Grid cells predict bound boxes B with a confidence point for each box that tells us how reliable the model is for that box containing a pattern and the box's accuracy in terms of pattern coverage. Therefore, each box gives 5 prediction values - x, y, w, h and the confidence point. Hence, the grid gives B such predictions. The grid also gives C the conditional layer probabilities, assuming that a pattern is present and the likelihood belongs to which layer. Only one set of layer probabilities is predicted per grid cell - regardless of the number of bounding boxes present. With "you only look one", YOLO has achieved very good processing speed (See Table 1). However, it is difficult to detect small patterns.

The SSD model was launched in 2016 by Liu et al. [23]. At the core of the SSD is the prediction of the category score and the bounding box number for a fixed set of bounding boxes using the small convolution filters applied to feature maps. To achieve high detection accuracy, predictions are generated at different scales from pattern maps at different scales and clearly separate predictions according to the aspect ratio. These design features make end-to-end training simple and high-precision, even on low-resolution input images, further improving the balance between speed and accuracy.

In 2017, new improvements to YOLO and SSD were introduced in YOLOv2 and DSSD, although there was an increase in accuracy, but the processing speed was slower.

See the results of Table 1, YOLO and SSD are suitable for real-time video pattern detection with FPS ≤ 45.
2.3. Face recognition

Human face recognition is a different problem than general pattern recognition problems. Because the human face has a different structure in the position of the eyes, nose, mouth, eyelashes, and ears, there are separate algorithms to solve this problem.

In 2000, Viola and Jones [27] launched a multi-layered training network, based on Haar-like feature of light/dark area difference between facial components, combined with AdaBoots algorithm achieved get good performance in real-time matching face detection. However, there are many limitations when the face is obscured or the head is tilted, face up/down.

There have been many groups of authors studying face detection algorithms. In 2016, Zhang et al [28] proposed a multi-tasking, deep layered model called MTCNN [28]. Firstly, the candidate window is created through the quick proposal network (P-Net). Secondly, the candidate areas in the next phase are refined through network refinement (R-Net). Thirdly, the output network (O-Net) creates the final bounding box and the position of the main landmarks of the face (Figures 2 and 3). Therefore, the accuracy is ≥ 82.1% high, maybe 95.9%, and the max FPS is 100 with GPU.

3. Suggest methods of pattern detection in video

3.1. Colour-based pattern detection

The problem is divided into two phases: 1- Value of threshold of colours is identified to detect pattern. 2- The colour area is detected within the threshold in the image (Figure 4).

Phase 1: The colour threshold can be predefined by entering the value into a given text file or when playing a video, the screen can be paused and the mouse is used to dot the pixels on the image to get the
RGB colour values. Then it is converted to the HSV colour system value. The upper or lower threshold value will be automatically calculated and saved in the text file.

Phase 2: Detecting colour areas in video: The video is played from files or directly from the camera. The frame is proceeded to read. Each frame is converted to HSV colour system. Then that each frame is filter out areas of colour that satisfy a given threshold. The methods of enlargement and contraction are carried out to smooth out the colour area. The contours are defined surround the coloured areas. The surrounding frame is drawn. Results such as: time, frame position, colour values and summary values can be produced after the end of the video.

![Diagram of pattern detection in video based on colour features](image)

**Figure 4.** Diagram of pattern detection in video based on colour features.

Results such as: Time, frame position, colour values and summary values can be produced after the end of the video. Experimental results are shown as bellow (Figure 5).

![The results when running the program to detect patterns in video based on colour features](image)

**Figure 5.** The results when running the program to detect patterns in video based on colour features.

### 3.2. Method of pattern detection based on shape and movement

A moving pattern in a video is essentially a change in its position against the background in each of the consecutive frames projected in the video. Therefore, the difference in the image area where the pattern takes its place in the frame is an indication of the movement of the pattern. See also [29].

Based on this feature, calculating the absolute difference value between two consecutive frames if greater than a certain threshold, it shows the movement of the pattern. Contour around the difference
will take the shape of the pattern. The position where the pattern appears is described by the rectangle surrounding the contour (Figure 6).

Figure 6. Diagram of method of pattern detection based on movement.

The program detects moving pattern in video by calculating the difference in pixel’s value between consecutive frames. Moving patterns will be marked with a rectangle surrounding them. The number of rectangles received is the number of motion detected (Figure 7).

Figure 7. The results when running the program to detect patterns in video based on movement.

The program is written in Python using the OpenCV library. Powered by Intel® Core ™ Processor (i5-3317U) 1.7GHz, RAM: 6.00GB (5.9GB used), Windows 10 Pro 64bit. The experimental results are as follows (table 2).
Table 2. Test results of the program to detect moving patterns in video.

| No | Type of cameras | The number of patterns | Has noise | The number of frames | The number of moving patterns calculated according to theory | Number of samples can be counted when experimenting |
|----|----------------|------------------------|-----------|---------------------|------------------------------------------------------------|--------------------------------------------------|
| 1  | Fixed          | 2                      | No        | 3428                | 6458                                                       | 4616                                             |
| 2  | Fixed          | 4                      | No        | 3292                | 13168                                                      | 8501                                             |
| 3  | Movement       | 3                      | Yes       | 303                 | 909                                                        | 4950                                             |

From the results of the experiment, it shows that the algorithm works well with fixed camera (background does not change or changes little).

During motion detection, the program displays the following information (time units in milliseconds, FPS - frames per second, ms). The following parameters are an illustration of a running program: FPS; Time for next frame; Calculated time; Delay time.

The end result shows the following information: Time Started; Time Ended; Path and File name; Shape of frames; Total frames; Total movements; Total calculation time (ms); Average computation time per frame (ms).

The execution time of the program depends on the quality of the video (FPS, height×width). With videos being archived files, the program works well because it is not real-time dependent.

With video from online cameras, the captured images in real-time. The program works well with cameras FPS = 30, when the resolution can be up to 1080×1920 pixels. For cameras FPS = 60, the resolution for the program to work well should be 720×1280 pixels or less.

3.3. Method of identifying human faces in video

Applying MTCNN and FaceNet model to detect human faces in photos, the author has adjusted the model to recognize human faces in the video. There are three phases: data preparation, training and prediction/identification.

**Stage 1. Data preparation:** The names or IDs of the people to be identified are used to name the folders, in which contain patterns of image of that person (there should be only one face in each image). Then the MTCNN network is used to crop out the area of the image containing only the face and resize of it to the same size.

**Stage 2. Training:** Adjusting the FaceNet network in the final “support vector machine” layer whose output label number corresponds to the name of the person folders (corresponding to the person's ID, or the number of output probability values is equal to the number of people to be identified). The result of this process will be a data file containing the volume values of the individual training process.

**Stage 3. Predicting for recognition the human face in the video:** The video is played from files or directly from the camera. For each frame, the image area is detected with the human face. Then it is put into the identity model, if the value is exactly ≥ a given threshold, the system will print out the person's name or ID. If it is not in database, the system will present "Unknown". Results are shown in figure 8.
Figure 8. Results of prediction for recognition the human face in the video.

4. Conclusion
Pattern detection in video is a problem with many real world applications. However, with video, the image quality of each frame and the presentation rate (FPS) will have great constraints on the algorithms. This article outlines the history of image pattern detection algorithms and evaluates their pros and cons. Various methods have been suggested for pattern detection in video. Among them, the colour-based pattern detection method works well when the pattern has unchanged colour and differs from background colour. The motion-based pattern detection method works well with video from fixed cameras. Human face recognition method, using MTCNN model network and reconfiguring the FaceNet model network is a method for specific problems. This method is suitable for image quality and real-time FPS speed in video.

References
[1] Hubel D H, Wiesel T N 1959 J Physiol 148 574–591
[2] Lawrence Gilman Roberts 1963 Machine perception of three-dimensional solids. Requirement for the degree of Doctor of Philosophy. Massachusetts Institute of Technology.
[3] David M, Shimon U, Tomaso A P 2010 Vision: A computational investigation into the human representation and processing of visual information (London: The MIT Press) p 11
[4] Martin A F, Robert A E 1973 IEEE transactions on computer 22 67–92
[5] Brooks R, Russell C, Binford T 1979 Proceeding of 6th Int’l Joint Conf. Artificial Intelligence 105–113
[6] David G L 1987 Artif. Intell. Mathematics, Computer Science 31 355–395
[7] Jianbo Shi, Jitendra Malik 1997 Normalized Cuts and Image Segmentation. Available at: http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Shi-Malik-CVPR-1997.pdf
[8] Lowe D G 1999 Proceedings of the International Conference on Computer Vision 1150–1157 doi:10.1109/ICCV.1999.790410
[9] Lienhart R, Maydt J 2002 Proceedings of the International Conference on Image Processing. doi:10.1109/ICIP.2002.1038171
[10] LeCun Y, Bottou L, Bengio Y, Haffner P 2013 Proceedings of the IEEE 86 2278–2324. doi:10.1109/5.726791
[11] Fukushima K 1980 Biological Cybernetics 36 193–202. doi: 10.1007/BF00344251.
[12] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do and Kaori Togash 2018. Insights into
[13] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T 2014 *Proceedings of the 22nd ACM international conference on Multimedia* 675–678 doi: 10.1145/2647868.2654889
[14] He K, Zhang X, Ren S, Sun J 2015 *Trans. Pattern Anal. Mach. Intell.* 37 1904–1916
[15] Girshick R 2015 *Proceedings of the IEEE International Conference on Computer Vision* doi: 10.1109/ICCV.2015.169
[16] Ren S, He K, Girshick R, Sun J 2015 *Proceedings of the 29th Conference on Neural Information Processing Systems* 91–99
[17] Jifeng D, Yi L, Kaiming H, Jian S 2016 *Proceedings of the 29th Conference on Neural Information Processing Systems* 379–387
[18] Tsung-Yi L, Piotr D, Ross G, Kaiming H, Bharath H, Serge B 2017 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* doi: 10.1109/CVPR.2017.106
[19] He K, Gkioxari G, Dollar P, Girshick R 2017 *Proceedings of the IEEE International Conference on Computer Vision* doi: 10.1109/ICCV.2017.322
[20] Zhong-Qiu Z, Shou-tao X, Xindong W 2019 *Proceedings of the IEEE Transactions On Neural Networks And Learning Systems* 30 3212–3232 doi: 10.1109/TNNLS.2018.2876865
[21] Everingham M, Van Gool L, Williams C K I, Winn J 2010 *Journal of Computer Vision* 88 303–338.
[22] Joseph R, Santosh D, Ross G, Ali F 2016 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* doi: 10.1109/CVPR.2016.91
[23] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C 2016 In: ECCV 2016. Lecture Notes in Computer Science (Cham: Springer) 9905 doi: 10.1007/978-3-319-46448-0_2
[24] Erhan D, Szegedy C, Toshev A, Anguelov D 2014 *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition* doi: 10.1109/CVPR.2014.276.
[25] Yoo D, Park S, Lee J-Y, Paek A S, Kweon I S 2015 *Proceeding of the IEEE International Conference on Computer Vision* doi: 10.1109/ICCV.2015.305.
[26] Najibi M, Rastegari M, Davis L S 2016 *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition* doi: 10.1109/CVPR.2016.260.
[27] Viola P and Jones M 2004 *Int. J. of Comput. Vision* 57 137–154
[28] Zhang K, Zhang Z, Li Z, Qiao Y 2016 *IEEE Signal Process. Lett.* 23 1499–1503
[29] Nguyen T C, Shasherev D 2018 *MATEC Web of Conferences* 155 01016. doi: 10.1051/matecconf/201815501016.