Detection of hijab syar'i as smart clothes system for moslem people using high performance of parallel computing

Detection of hijab syar'i as smart clothes system for moslem people using high performance of parallel computing

I Cholissodin[1,*], D E Palupi[1], M Y Y Putra[1] and S Aprilisia[1]

[1]Faculty of Computer Science, Brawijaya University Jl. Veteran Malang, Jawa Timur 65145, Indonesia.

[*]E-mail: *imamcs@ub.ac.id*

Abstract. Moslems are required to remain consistent in wearing syar'i clothing. Sometimes, however, it is very difficult to distinguish between syar'i clothes and the non-syar'i ones, especially in the current millennial era and until the end of time there are many temptations that are not good. Digital image processing in the present day has become one of the areas that can be processed to help various problems associated with images. However, it requires very long processing time and ineffective processing. The CPU alone is not enough. Therefore, it takes processing time allocation that can cut the time to speed up the process. This study utilizes the way to speed up performance analysis of digital images through GPU. The performance analysis of GPU includes CUDA and Yolo because it allows specialists in parallel programming to use GPU resources and can perform the object detection process quickly and accurately by applying an artificial neural network to the image of someone who is wearing syar'i hijab as smart clothes for Moslem women. Several hardware, software, and file dependency specifications are employed to support the implementation process. The test results achieve the accuracy of 100%, proving that by using the implementation of Yolo under GPU.

1. Introduction

Today, the development of computer technology is growing, causing the produced, stored and accessed digital images to be more in number and more complex. Digital image is currently used in various fields such as security, commerce, and many more uses of information in the form of images in other fields. Digital image itself is an image generated from discrete signals. The bit depth in a digital image can affect the light intensity value. The image is formed by a group of picture elements where there are two information, which are pixel coordinates (x,y) and pixel intensity values $f(x,y)$. There are several ways that can be utilized to store digital images in memory. The determination of a type of digital image that has been formed can be seen from the way the storage process is. Binary image, Grayscale image, and color image are the most commonly used image types. Processing an image can be performed by improving the quality of existing images, but the growing world of computation is characterized by the increased capacity and speed of computer processes as well as the emergence of computational sciences that enable humans to retrieve information from an image [1,2].

Identification and analysis of human clothing is an interesting issue and continues to be developed from year to year associated with many applications that have very complex structures and have their respective characteristics. From these unique characteristics, research shows that there is a wealth of information that can be obtained from the analysis of a person's clothing image such as expressions,

preferences, emotions and so on. Therefore, there are various topics that attract attention and are studied intensively. The field of computer vision itself continues to strive for ways to make computers able to imitate the way the human senses work. In relation to this issue, creating a system that can detect Moslem women who wear syar'i hijab becomes more interesting [3]. The reason is that sometimes a Moslem woman does not have much understanding about hijab that are syar'i and those that are not syar'i. One of the information contained in headdress clothing is the use of the veil or others as a hijab. A woman is wearing the hijab as a symbol of obedience to the rules of Islam and an easy recognition that she is a Moslem. To help overcome the detection problem, a system is required to assist Moslem women who do not know the syar'i hijab through the use of camera photos that have been processed with the intelligence system algorithm. The approach used in the beginning utilizes the information of a person who is wearing hijab by employing CUDA, which is one of the architectures for the users to do image processing that utilizes the GPU openly. Image processing can be processed in parallel using the GPU, where initially, each pixel in the image that is processed sequentially, i.e. one by one in an order, can be processed in parallel at the same time for several pixels in one process by dividing the process into the cores on the GPU, with an algorithm to detect the objects using Yolo Darknet [4].

2. Methods

2.1. Hijab

The meaning of Hijab is in accordance with the one mentioned in the Qur'an Surah al-Ahzab verse 53 that ordered the wives of the Prophet not to talk to other people besides a mahram except with the curtain barrier in a direct contact, or by closing their face. The meaning of the verse is: "... And when you ask for something (needs) from them (the wives of the Prophet Muhammad SAW), ask them from behind a partition/curtain. This way is purer for your heart and their hearts..." This verse is mentioned related to ethical issues when one was together with the wives of the Prophet. Therefore, hijab, in accordance with the verse above, refers to a cover for direct contact, not only related to clothing. Meanwhile, what about the hijab, veil or the other. Hijab is mentioned once in the Qur'an, in Surah al-Ahzab verse 59 in the form of plural, *jalabîb*. In the Verse, it means, "O Prophet, tell your wives, your daughters, and the women of the believers (that) they must bring down over themselves parts of their *jalabîb* (outer garments). That is more suitable that they will be known, and not be abused. Allah SWT is the Most Forgiving and the Most Merciful." These verses provide lessons and wisdom on the importance of wearing the hijab as a cover, clothing and others. These verses also provide an explanation related to the characteristics of the syar'i hijab [5].

Figure 1. Contemporary (but refer more to as non-syar'i) hijab vs syar'i hijab.

The difference of contemporary hijab versus syar'i hijab according to Figure 1 shows in this case that contemporary hijab for example is more indicated as non-syar'i hijab although it does not always mean that contemporary hijabs are not syar'i. It means that there are also syar'i contemporary hijabs (such as the term of fashionable syar'i hijab which makes many Moslem girls, teenagers or women wear the hijab). Some of the characteristics of non-syar'i contemporary hijab are being ignorant to visible forehead hair, wearing striking accessories such as earrings, necklaces and contact lenses, wearing heavy makeup such as striking use of eyebrow pencil, mascara and blush on, wearing veil that does not cover the chest, wearing clothes and pants of tight (curvy overfitting clothes and pants), wearing clothes that do not cover the entire *aurat* (forbidden parts to show) such as parts of the arms, feet, and carrying bags or other excessive items. On the other hand, the syar'i hijab is not superfluous but still elegant, including when using face powder or blush on, lip moisturizer and others, wearing a wide veil covering the chest, wearing a backpack, and covering the whole body except the face, palms and hands [6,7].

2.2. GPU Programming and CUDA

GPU (Graphical Processing Unit) is a processor that has a special function for rendering on graphics cards only. Along with the increasing needs of rendering, however, comparable to the capabilities of the graphics processor, the accelerated increase in GPU technology is faster than the increase in the actual processor/CPU technology, and eventually GPU becomes General Purpose, meaning that it is no longer used only for rendering but can also be used for general computation.

CUDA is an API developed by Nvidia that is used to do a computation that can run parallelly or in other words together. Developers can use CUDA for general purpose processing called the GPGPU (General-purpose computation on GPU) approach. CUDA platform is an access layer to software that directly provides the virtual GPU instruction set for the kernel as the executor of the calculation. CUDA platform is designed to work with programming languages such as C, C++, and Fortran. This accessibility makes it easier for specialists in parallel programming to use GPU resources, which is in contrast to previous APIs such as Direct3D and OpenGL, which require advanced skills in graphical programming. When it was first introduced by Nvidia, the name CUDA was an acronym for Compute Unified Device Architecture [8].

2.3. Yolo Darknet

Yolo Darknet is a framework that can be implemented on a CPU or GPU developed by Joseph Chat Redmon. Yolo itself is a feature of darknet to detect objects very quickly and accurately by applying artificial neural networks to images. Then, the network groups the images into parts and predicts parts of the objects and possible edges of the objects in each part. The detection speed is determined by *the hardware* used, which generally shows that the detection using the GPU is faster than using the CPU by up to 20 fold. In Yolo v3, there is a change in Loss Function, where previously, the Yolo v2 loss function appeared as shown in Equation 1.

$$\begin{aligned}
&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\\
&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\\
&+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2\\
&+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2\\
&+\sum_{i=0}^{S^2}1_i^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2
\end{aligned}\tag{1}$$

One of the modifications of YOLO v2 is squared errors (SE), which in YOLO v3, it has been replaced with the term cross-entropy error. In other words, object confidence and class predictions in YOLO v3 are currently predicted through logistic regression [9,10,11].

3. Design and Implementation

In this research, the design and implementation is related to Yolo v3 configuration files and GPU devices used by utilizing Google Colaboratory with the specifications of HDD ± 300GB RAM 12GB and GPU NVIDIA Tesla K80 type on Intel(R) Xeon(R) CPU @ 2.30GHz with cudnn-10.0-linux-x64-v7.5.0.56.tgz with Ubuntu 18.04.3 LTS. The results of the program code implementation and its resources can be observed at https://github.com/imamcs19/GreenTech2019-Hijab-Detection. In running Yolo Darknet, it is initiated by conducting a training on the data that will be used.

____*Stages of image data processing for syar'i hijab detection*____

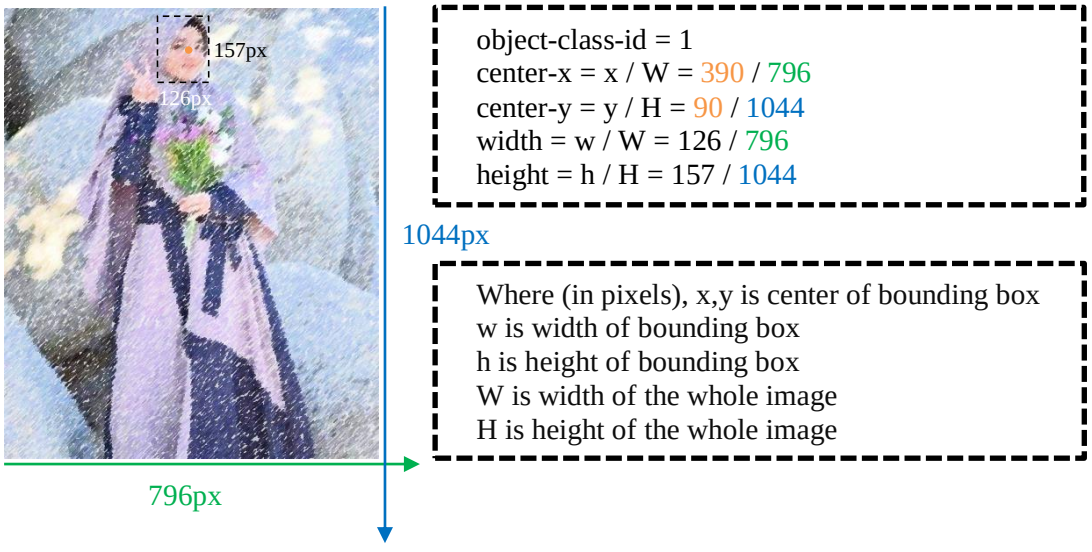
1. For preprocessing
   - **a.** An initial data acquisition using *google_images_download* with Jupyter Notebook via the following command

     ```
     !googleimagesdownload --keywords "hijab syari" --limit 500 –-chromedriver \
     C:\..\chromedriver.exe
     ```

     where the location of file "chromedriver.exe" is obtained with the command of "C:\Users\..>where /r C:\chromedriver.exe" in cmd and

     ```
     !googleimagesdownload --keywords "foto suami istri hijab" --limit 500 \
     --chromedriver C:\..\chromedriver.exe
     ```

     for example the number of data to crawl is 500
   - **b.** Deleting the data of non-valid-crawled image
   - **c.** Renaming image data with a sequence from small to large numbers
2. Data processing using Yolo v3 algorithm
   The steps in this process are established by conducting a single classification (only one class), which is the way the system is able to recognize images in which there are women who wear the hijab only.
   - **a.** Data Annotation using tool to create Bounding Box to train process easily
     <object-class-id> <center-x> <center-y> <width> <height>
     The last four entries of data annotation are set as floating-point values between 0 to 1.



b. Execute this script on your Google Colabolatory to training process (Part 1 of 2)

```
1  from google.colab import drive
2  drive.mount('/content/drive')
3  import os
4  os.chdir("/content/drive/My Drive/darknet/cuDNN")
5
6  #download cuDNN form thrid-party Nvidia, save on google drive
7  !wget http://people.cs.uchicago.edu/~kauffman/nvidia/cudnn \
8  /cudnn-10.0-linux-x64-v7.5.0.56.tgz
9
10 # cudnn-10.0-linux-x6 100%[================>] 412.76M  13.3MB/s    in 43s
11
12 # 2019-09-14 15:17:13 (9.60 MB/s) - ‘cudnn-10.0-linux-x64-v7.5.0.56.tgz’ saved
13 # [432811879/432811879]
14
15 # Extracts cuDNN files from Drive folder to the VM CUDA folders
16 !tar -xzvf ./cudnn-10.0-linux-x64-v7.5.0.56.tgz -C /usr/local/
17 !chmod a+r /usr/local/cuda/include/cudnn.h
18
19 # Now we check the version we already installed (optional)
20 !cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2
21
22 # run only once
23 os.chdir("/content/gdrive/My Drive/git")
24
25 # Comment this code on the future runs (cause run only once)
26 !git clone https://github.com/kriyeng/darknet/
27 %cd darknet
28
29 # Run only once
30 !git checkout feature/google-colab
31
32 #Compile Darknet (run only once)
33 !make
34
35 #Change directory
36 os.chdir("/content/drive/My Drive/darknet/weights")
37 !wget https://pjreddie.com/media/files/darknet53.conv.74
38 # darknet53.conv.74   100%[================>] 154.96M  17.3MB/s    in 10s
39
40 os.chdir("/")
41 !mkdir mydarknet
42 %cd mydarknet/
43
44 # Copy the Darkent compiled version to the VM local drive
45 !cp /content/drive/My\ Drive/darknet/bin/* ./darknet
46
47 !chmod +x ./darknet
48 !chmod -R 755 /content/drive/My\ Drive/darknet
```

c. After Part 1 is completed, run this script to complete the training process (Part 2 of 2)

```
!./darknet detector train "/content/drive/My Drive/darknet/obj.data" \ "/content/drive/My
Drive/darknet/cfg/yolov3.cfg" "/content/drive/My \
Drive/darknet/weights/darknet53.conv.74" -dont_show
```

If Error: l.outputs == params.inputs appear when you are using yolo v3, this error occurs when your filter value is wrong in any of the convolution layer before [yolo] layer. It should be = num/3*(classes+5). If you are using yolo v2, this error happens when your filter value in convolution layer before region layer is wrong. It should be = num*(classes+5).

d. Performing the testing process

```
os.chdir("/content/drive/My Drive/darknet")
!tar -xzvf ./cuDNN/cudnn-10.0-linux-x64-v7.5.0.56.tgz -C /usr/local/
!chmod a+r /usr/local/cuda/include/cudnn.h

os.chdir("/")
!mkdir mydarknet

# Copy the Darkent compiled version to the VM local drive
!cp /content/drive/My\ Drive/darknet/bin/* ./mydarknet/darknet
!chmod +x ./mydarknet/darknet
os.chdir("/mydarknet")

# Configure labels
!mkdir -p ./data
!cp -R /content/drive/My\ Drive/darknet/labels ./data

!./darknet detector test "/content/drive/My Drive/darknet/obj.data" \
"/content/drive/My Drive/darknet/cfg/yolov3.cfg" "/content/drive/My \
Drive/darknet/backup/yolov3_2000.weights" "/content/drive/My \
Drive/darknet/testImgFamily/f1.jpg" -thresh 0.85 -dont-show

# Show the result using the helper imgShow()
imShow('predictions.jpg')
--------------
layer     filters    size              input                output
0 conv     32  3 x 3 / 1   416 x 416 x   3   ->   416 x 416 x  32 0.299 BF
1 conv     64  3 x 3 / 2   416 x 416 x  32   ->   208 x 208 x  64 1.595 BF
   ..

Total BFLOPS 65.290
Allocate additional workspace_size = 52.43 MB
Loading weights from
/content/drive/My Drive/darknet/backup/yolov3_2000.weights...
seen 64
Done!
/content/drive/My Drive/darknet/testImgFamily/f1.jpg: Predicted in 40.750000 ms.
hijab: 90%
```

4. Results and Discussion

4.1. Testing with Variation data

This research is limited to the detection of hijab in parts of the face only, not on the detection of the veil that cover the entire body which includes *aurat* (body parts required by Islam to be covered). Therefore, the concept of syar'i here is still limited by taking the image data that contain a woman who cover her whole *aurat* with big hijab and other clothing that are completely closed and not too tight.

Figures 2, 3 and 4 show that the detection of syar'i hijab has been tested on a number of quite complex variations of photos. Although it has not been directed to distinguish whether the hijab is really syar'i or not, the system built has been successful to detect the hijab worn by a Moslem, even though part of the bounding box is still partly in the face. The output is able to detect the hijab not by detecting someone's face because in this study the focus is not on face detection, but the detection of hijab.

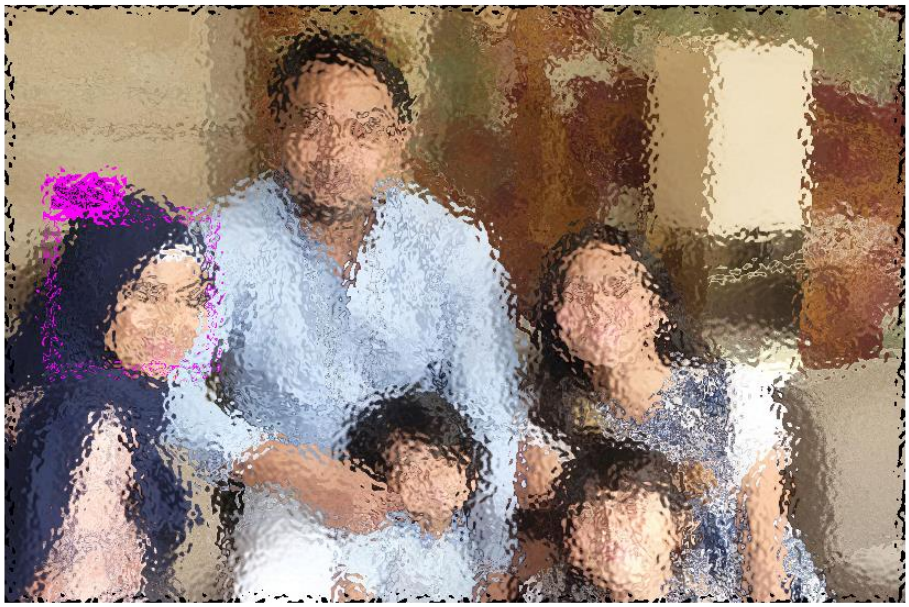


Figure 2. The result of the detection of a woman wearing a hijab in a family photo with complex conditions, where some members of the family are wearing no hijab (the husband) and their young children.

Figure 3. Hijab detection result worn by a mother carrying her child next to her husband.

Figure 4. Hijab detection results for a Moslem woman in her own picture.

Figure 5. Results of syar'i hijab detection on other test data

In Figure 5, the data used for testing has two specifications. First, the image of a Moslem woman taking a single photo, and the second is a collection of images of two Moslem women taking a picture together. One of the things that causes system difficulties in identifying the hijab is the position of a Moslem woman. Besides that, it is also due to the high variety of hijab shapes, ranging from the size, color, to the model of the hijab itself. Moreover, the face shape of each Moslem woman is also varied, especially when there is additional makeup or decoration on the face.

Another difficulty occurs when some of the Moslem woman's fingers are placed near the face and the hijab worn. The overall results of the testing of image data of Moslem women wearing hijab have been able to show that the performance of the system built is quite reliable. The main note to review regarding the application key to the success of testing the algorithm of Yolo v3 is the availability of data, especially for the training should be available in plenty of numbers. If data are lacking, then the system will be very vulnerable to come to the underfitting condition, where the system is experiencing less than optimal during the training process in reading and modeling the object patterns. Consequently, this will make the system premature to carry out the testing process

4.2. Testing train Loss

In Table 1, the accuracy results of the data shown are shown. The accuracy results prove that the Yolo v3 algorithm has quite good performance although sometimes there are some hijab data that have not been recognized accurately. This is identified because it is necessary to add as many training data as possible so that the system built is able to learn optimally.

Table 1. Result of data variation test.

Data Test	Accuracy (%)
Individual or Groups 1	100%
Individual or Groups 2	100%
Individual or Groups 3	100%
Individual or Groups ..	100%
Individual or Groups 11	100%
Average Accuracy	100%

Based on Figure 6 during the Yolo training process, the log data displayed contain one of which is the loss value that can also be called as the magnitude of the error rate. The training process can be considered ideal if the loss value is increasingly converging toward zero (train 1st achieve 0.328299, and train 2nd achieve 0.110182) together with the increasing number of iterations or Batch Number. This Loss Value can be compared to a value of epsilon on a specified amount as the algorithm of artificial neural network (ANN) are generally in each iteration. The epsilon value must also go through trial-and-error several times in order that the obtained results of the detection of the desired object is optimal because each case has a different complexity of pattern recognition objects.

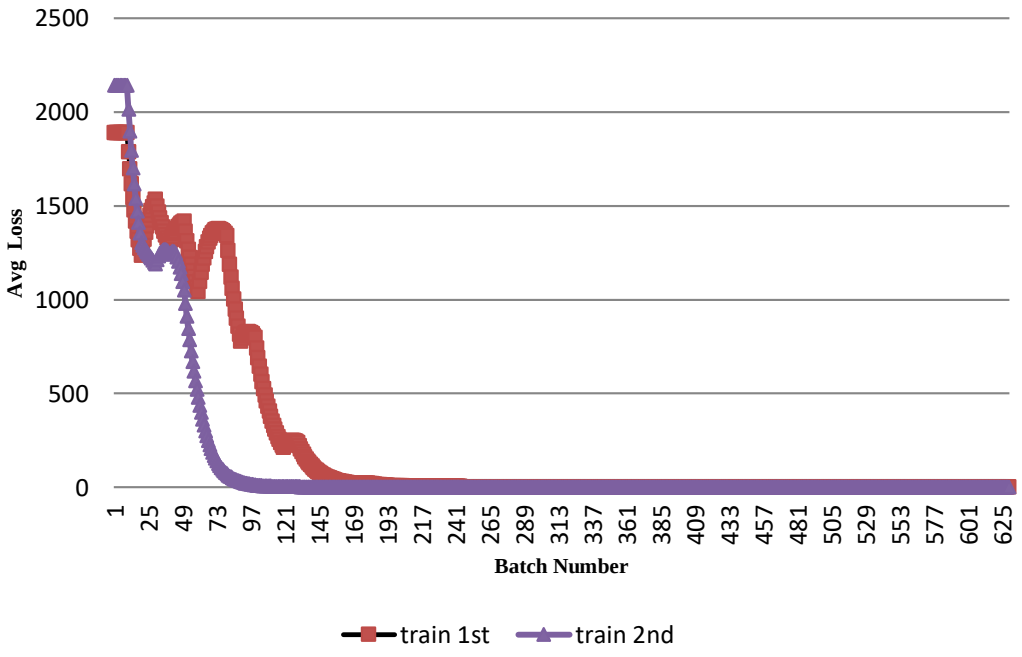


Figure 6. The best average value of Loss Function.

5. Conclusion

Based on the results of research on performance analysis of digital images using GPU, it can be concluded that processing the types of digital images is more powerful by utilizing GPU, the research implementation of which is using NVIDIA CUDA. For the digital image processing performance for hijab detection based on average train loss data, the graph leads to convergent conditions with a quite optimal accuracy value of 100% for data train as data test adds few others new relevant data as testing process. Then, the suggestion for research in the future is to prepare several more complex data images, for instance by giving some classes or categories of syar'i to non-syar'i hijab with their rating assessment as well. The main constraint comes from how to make the selected data used remains the ones that are really syar'i. One of the ways is to carry out the training process by collecting various kinds of shapes of syar'i hijab images from the types of hijab, clothing and others in a variety of conditions, for example the angle and position of the image. In addition, in the next research, it is expected that a detection system that is equipped with auto-augmented hijab for both Moslem women and men will be developed. Furthermore, this hijab processing would be better to be applied in real time using a mobile application that is able to utilize the GPU on the device or based on cloud computing. This is because the millennial generation today uses mobile phones in their daily activities. Additionally, we recommend the use of better Yolo methods, for example by Yolo Pyramid Spatial Pooling (SPP) or the other methods and optimization YOLO v2 parameters λ_{coord} and λ_{noobj} using Particle Swarm Optimization (PSO) or Genetic Algorithm (GA) or other algorithm from Swarm Intelligence and Evolution Algorithm.

References

[1] Gonzalez R C and Woods R E 2018 *Digital Image Processing, 4Th Edition* (USA: University of Tennase)

[2] Koprawi M, Teguh B A and Dani A 2017 *Analisis Peforma Komputasi Paralel GPU Menggunakan PYCUDA dan PYOPENCL dengan Komputasi Serial CPU pada Cita Digital* (Yogyakarta: Teknik Elektro & Teknologi Informasi Universitas Gadjah Mada)

[3] Khaliluzzaman M, Begum A, Fatama S K and Islam M M 2017 *20th International Conference of Computer and Information Technology (ICCIT)* (Dhaka: ICCIT Press) p 1-6

[4] Jupiyandi S, Saniputra, F R, Pratama Y, Dharmawan M R and Cholissodin I 2019 *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)* **6** (4) 413-419

[5] Syauqi M I 2019 *Ini Perbedaan Istilah Jilbab dan Hijab dalam Syariat* (Jakarta: www.Islam.Nu.or.id) available at https://islam.nu.or.id/post/read/82291/ini-perbedaan-istilah-jilbab-dan-hijab-dalam-syariat on

[6] Fitri F 2019 *Ini Perbedaan Hijab Kekinian dan Hijab Syar'i, Cewek Muslim* (Jakarta: www.wowmenariknya.com) available at https://www.wowmenariknya.com/mozaik/ini-perbedaan-hijab-kekinian-dan-hijab-syari-cewek-muslim-masuk/ on 12 September 2019

[7] Al Hadiid FT, 2015 *HSD International Hijab Solidarity Day* (Malang: alhadiid.teknik.ub.ac.id) available at http://alhadiid.teknik.ub.ac.id/ihsd-international-hijab-solidarity-day/

[8] Cholissodin I, Maghfira T N 2017 *Pemrograman GPU* (Malang: Fakultas Ilmu Komputer, Universitas Brawijaya, Malang)

[9] Ibánez D 2019 *8 Tips for Google Colab Notebooks to Take Advantage of Their Free-of-charge 12GB-RAM GPU* (Barcelona: Dev) available at https://dev.to/kriyeng/8-tips-for-google-colab-notebooks-to-take-advantage-of-their-free-of-charge-12gb-ram-gpu-be4

[10] Nayak S 2019 *Training YOLOv3: Deep Learning based Custom Object Detector* (California: University of California) available at https://www.learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/

[11] Redmon J, Farhadi A 2016 *YOLO9000: Better, Faster, Stronger* (University of Washington: Allen Institute for AI) available at http://pjreddie.com/yolo9000/