Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Improved YOLOv5 with Backbone Replacement to MobileNet V3s for School Attribute Detection

Ardanu Dhuhri Nugroho[1)*], Wiga Maulana Baihaqi[2)]
[1,2)]Universitas Amikom Purwokerto, Indonesia
[1)]ardanuduhri@gmail.com, [2)] wiga@amikompurwokerto.ac.id

Submitted : Jul 14, 2023 | **Accepted** : Aug 2, 2023 | **Published** : Aug 3, 2023

Abstract: School attributes are a series of clothes and accessories that must be worn by students in the school environment. The implementation of this rule aims to create discipline in students. However, in practice, not all rules can be implemented properly because there are still students who violate these rules. One of the rules applied at school is the use of complete attributes. Currently, attribute checks in schools are done manually or through teacher supervision. However, this takes more time, is prone to errors, and is inefficient due to the large number of students being checked. This study proposes an improved YOLOv5 architecture with the replacement of the backbone to MobileNetV3s to detect school attributes. This method uses deep learning and the YOLOv5 algorithm to detect in real time the use of school attributes by students. In this study, the experimental results show that the enhanced YOLOv5 with MobileNetV3s has higher accuracy compared to the original YOLOv5. In addition, the improved model is more efficient in memory usage and weight file size. With an accuracy result of 0.912 on mAP50 and a weight size of about 90 MB and a memory usage of <7 GB, it shows the potential of replacing the backbone in this technology in overcoming attribute detection challenges in schools and can be applied in other cases. However, further research is needed to generalize these results to other problems. This research also shows that backbone replacement in YOLOv5 can affect the accuracy of the model.

Keywords: Backbone Replacement, Computer Vision, Deep Learning, MobileNetV3s, School Attribute, YOLOv5.

INTRODUCTION

Every school must have its own rules or regulations that must be obeyed and carried out by all people in the school environment. School rules have a binding and coercive nature. This is a marker of student behavior, where the purpose of making school rules is to create discipline in students. However, in reality, not all rules can be implemented properly because there are still students who commit violations (Komah & Budjang, 2017). To overcome this problem, supervision, and checking of the applicable rules must be carried out. One of the rules applied in schools is the use of complete attributes. Currently, attribute checking in schools is done manually or through teacher supervision. However, this can take more time, is prone to errors, and is inefficient due to the large number of students who are checked for attributes. Therefore, accuracy and assertiveness when checking must be improved. Moreover, on Monday where every school must hold a ceremony that must be attended by all students, and must wear complete school attributes. The ceremony itself usually consists of speeches, singing the

national anthem, and raising the national flag. The purpose of the flag ceremony is to foster a sense of love for the country and discipline in students starting from an early age. This is so that they are familiar with these values and can implement them in their daily lives, both in the school environment, family, and society (Audina et al., 2022). Therefore, this ceremony activity must be followed by students seriously in following the rules. Following established school policies, students who do not meet the requirements of school uniforms with accessories will be subject to disciplinary punishment (Prayitna et al., 2022).

The issues that have been described and coupled with increasingly sophisticated technological advances, especially in computer vision and deep learning have recently transformed various fields, including object detection and image classification. There are several deep learning algorithms for object detection and classification tasks including, Single Shot Detector (SSD), (Faster Recurrent Convolutional Neural Network) Faster R-CNN and You Only Look Once (YOLO) (Gelar Guntara, 2023). the YOLO model or algorithm showed the best performance. The faster RCNN model is the most superior among the RCNN variants, but the FPS is not satisfactory because it uses CNN. While the SSD is fast, it uses mobile-v1 and this model is lighter, resulting in lower accuracy in terms of precision (Kim et al., 2020). In this school attribute detection process using the YOLO algorithm. The YOLO version used is YOLOv5. YOLOv5 is known for its real-time object detection capabilities, making it a popular choice for applications that require efficient and accurate detection. The YOLO method detection system is proven to be faster and more precise for identifying objects in images or images so it is most suitable if used for real-time object detection on video (Iskandar Mulyana & Rofik, 2022). Some of the main advantages of YOLOv5 compared to previous versions are smaller size, higher speed, higher precision, and the use of the ecologically mature PyTorch open-source ML framework(Horvat & Gledec, 2022). However, when YOLOv5 is used for specialized detection tasks that require precision and there is a need for further improvement, it requires modifications to the architecture or parameters of the YOLOv5 algorithm itself. The most common thing to modify the architecture of YOLOv5 is to change the backbone of YOLOv5. Backbone is a convolutional neural network that is used to take input images and extract features from the input images (Horvat & Gledec, 2022). It is a crucial component in detecting any object, as it is the main framework in charge of pulling contextual information from the input image and transforming it into a more abstract pattern (Benjumea et al., 2021). This backbone replacement is done by replacing it with a popular CNN architecture. Examples of CNN architectures that are often used for backbone replacement include AlexNet, VGG16, VGG19, ResNet50, ResNet101, GoogleNet, Inception-V3, Inception-ResNetV2, and SqueezeNet (Setiawan, 2020). In this school attribute detection case, the architecture used to replace the original backbone of YOLOv5 with MobileNet V3s architecture. There is a reason for choosing this architecture, which is because this architecture is capable of producing high accuracy as well as light weight and can run on computer devices with not too high performance (personal computers or laptops) (Hastomo & dan Sudjiran, 2021). The MobileNetV3s backbone offers several advantages over the default backbone used in YOLOv5. First, MobileNetV3s is specifically designed to be lightweight, making it suitable for resource-constrained environments. This characteristic is particularly important in applications such as school attribute detection, where deployment on edge devices or low-power platforms may be required. Secondly, MobileNetV3s achieves a balance between computational efficiency and model accuracy. Its architecture incorporates various innovative design choices, such as efficient inverted residuals and squeeze and excitation modules, which contribute to improved performance. By integrating MobileNetV3s into YOLOv5, we can leverage these advances to improve detection accuracy while maintaining real-time performance.

In this paper, we propose an improved YOLOv5 architecture with backbone replacement to MobileNetV3s for school attribute detection. We demonstrate the effectiveness and efficiency of the integrated model through comprehensive experiments on data sets. The results highlight the potential of changing the backbone to this technology compared to the results of the original YOLOv5 architecture in addressing the specific challenges of attribute detection in schools, which may be implemented in other cases.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12702 p-ISSN : 2541-044X

LITERATURE REVIEW

YOLO, a unified model for object detection (Redmon et al., 2016). A deep learning-based object detection framework, which has revolutionized the field of object detection introduced in 2015. The original YOLO model can analyze images in real-time, processing 45 frames per second. A more compact version called Fast YOLO achieves an impressive speed of 155 frames per second while maintaining twice the average precision (mAP) of other real-time object detection systems (Redmon et al., 2016). And in 2016 it introduced the YOLO9000: Better, faster, stronger (Redmon & Farhadi, 2017). An advanced real-time object detection system that can detect more than 9000 categories of objects. These improvements are not limited to the use of Darknet-19 as a new architecture but also include batch normalization, higher input resolution, convolution layers with anchors, dimensional clustering, and fine-grained features. Using the new multi-scale training method, the same YOLOv2 model can run at multiple sizes, offering an easy trade-off between speed and accuracy. At 67 FPS, YOLOv2 achieved 76.8 mAP in VOC 2007. At 40 FPS, YOLOv2 achieved 78.6 mAP, outperforming cutting-edge methods such as Faster RCNN with ResNet and SSD, but still running significantly faster. YOLOv3: Incremental Improvement (Redmon & Farhadi, 2018). At 320×320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When looking at the old .5 IOU mAP detection metric, the YOLOv3 does quite well. YOLOv3 achieved 57.9 AP50 in 51 ms on the Titan X, compared to 57.5 AP50 in 198 ms by RetinaNet, the same performance but 3.8 times faster. YOLOv3 is a good detector. It's fast and accurate. It's not very good at the COCO average AP between 0.5 and 0.95 IOU metric. But it is very good on old detection metrics of 0.5 IOU. YOLOv4: Optimal Speed and Accuracy of Object Detection (Bochkovskiy et al., 2020). Tons of features are said to improve the accuracy of the Convolutional Neural Network (CNN). Some features operate on certain models exclusively and for certain problems exclusively, or only for small-scale datasets; while some features, such as batch-normalization and residual-connection, can be applied to most models, tasks, and datasets. New features in YOLOv4: WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss, and combine some of these features to achieve state-of-the-art results: 43.5% AP (65.7% AP50) for the MS COCO dataset with a realtime speed of ~65 FPS on a Tesla V100. For YOLOv5 there is no exact source of who created it from the research paper, but it is the first version of YOLO implemented on Pytorch, not Darknet. YOLOv5, like YOLOv4, uses CSPDarknet53 as its architectural backbone (Jocher et al., 2020). The launch consists of five different model sizes: YOLOv5s (smallest), YOLOv5m, YOLOv5l, and YOLOv5x (largest). One of the main improvements in the YOLOv5 structure is the incorporation of the Focus module, denoted by the solitary module, which is formed by replacing the initial three modules of the YOLOv3. This merger reduces the number of modules and parameters while increasing the forward and backward propagation speeds, all without significantly affecting the mean average precision (mAP). There are several other versions of YOLO itself, but the most modification of the YOLO architecture is done in the YOLOv5 version by replacing the backbone part with a Convolutional Neural Network (CNN) architecture. With the implementation of PyTorch for YOLOv5, it will be easier to replace the backbone or merge it with the Convolutional Neural Network (CNN) architecture. YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles(Benjumea et al., 2021). who proposed a series of models at different scales, named 'YOLO-Z', and who displayed up to 6.9% improvement in mAP when detecting smaller objects at 50% IOU, at the cost of only 3ms improvement in inference time compared to the original YOLOv5. YOLO-Z has experimented with replacing the existing backbone in YOLOv5 with ResNet and DenseNet as the backbone. For the backbone replacement itself, you can use popular CNN architectures that support PyTorch, one of which is MobileNet. MobileNet is an efficient CNN architecture model designed with 2 sets of hyper-parameters to build a very small and low latency model that will be easily implemented according to the needs of mobile and embedded applications(SUYUTI, 2023). MobileNetV3 is divided into two variants, MobileNetV3-Large and MobileNetV3-Small, which are designed for scenarios with high and low resource requirements and these models are then adapted and applied to object detection and semantic segmentation tasks (Howard et al., 2019). MobileNetV3 variants achieve more advanced results for mobile classification, detection, and segmentation. MobileNetV3-Large is 3.2% more accurate in ImageNet classification while reducing latency by 20% compared to MobileNetV2. MobileNetV3-Small is 6.6% more accurate compared to the MobileNetV2 model with comparable

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

latency. MobileNetV3-Large detection is more than 25% faster with approximately the same accuracy as MobileNetV2 on COCO detection. MobileNetV3-Large LRASPP is 34% faster than MobileNetV2 R-ASPP with similar accuracy for segmenting Cityscapes(Howard et al., 2019).

METHOD

The research flow used for this research can be seen in Figure 1, which illustrates the stages of research. It consists of five main processes that will be carried out, namely: Dataset Preparation, Backbone Changing, Modelling, Training Model, and Evaluation Performance.

Figure 1. Workflow Research

Dataset Preparation

The dataset for this research was obtained from image data taken from elementary schools, and junior high schools, and some images from the internet with a total of 334 images.. From the image data, bounding box, preprocessing, and augmentation using roboflow web are performed. Furthermore, from the dataset, there are four classes, with details of 255 hats, 219 ties, 211 belts, and 184 bets. Figure 2 presents a representation of the dataset used in this study.

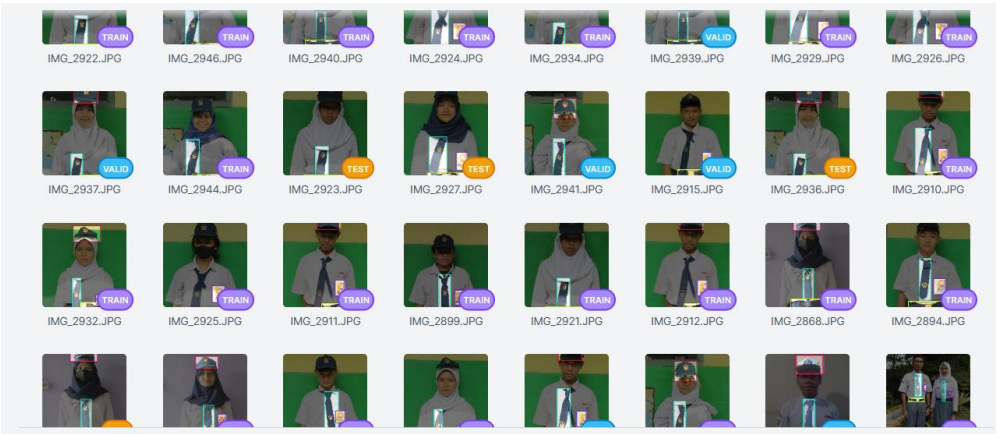


Figure 2. School Attribute Dataset

Backbone Changing

Backbone changes were made in this study, changes using the MobileNet V3 Small CNN architecture. Mobile Netv3 inherits the strengths of the previous version, introduces several new modules, and uses the latest Neural Architecture Search (NAS) technology to automatically find the optimal neural network structure on the target data set. Mobile Netv3 introduces a channel attention mechanism, called the SE module, to improve model performance. The SE module aims to learn the correlation between channels and give higher weights to more important channels, thus increasing their significance in the model. This mechanism improves the overall performance of the model. In addition, MobileNetv3 uses NAS (Neural Architecture Search) technology to optimize the network structure. Through this process, some layers that require long computation time are optimized or eliminated. For example, the number of convolution kernels in the first convolution layer was reduced from 32 to 16. Surprisingly, the accuracy of the network remained almost constant despite the reduction in layers.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 3. MobileNet V3 small architecture

From the architectural drawing in Figure 3, Mobile Netv3 is broadly divided into feature, avgpool, and classify, and their functions are feature extraction, global pooling, and classifier. of all layers, only 13 layers are taken as feature extractors, and the last three downsampling parts. Furthermore, it will be subdivided into three parts that start and move forward from the end. Bneck itself is a block of layers owned by mobilenetv3. Each block consists of a narrow input and output (bottleneck), which has no nonlinearity, followed by expansion to a higher space and projection to the output (HOWARDS). Figure 4 illustrates part of the bottleneck or Inverted Residual layer of mobilenetv3.

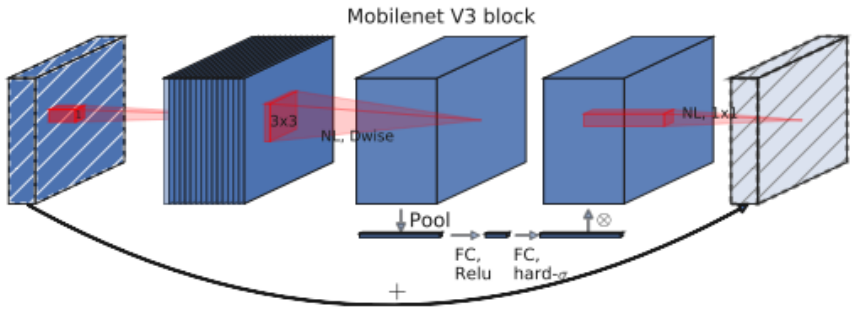


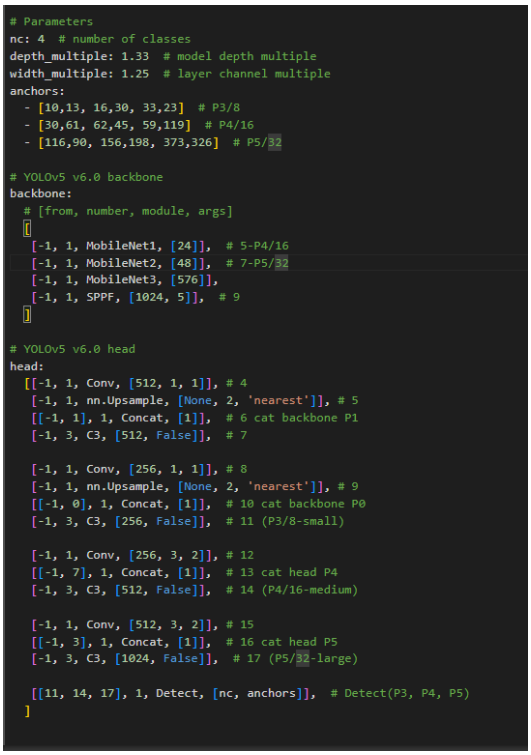
Figure 4. Mobilenet V3 Block

The 13 layers taken will be divided into three parts, namely, the first part, layers one to four, the second part, layers five to nine, and the last part is taken from layers 10 to 13.

Modelling

After getting a new backbone from MobileNet V3 Small, the next step is to create a model. In making a model in YOLOv5 there is a YAML config file that must be adjusted to the new backbone. This YAML config file consists of a section, backbone, and head, Figure 5 is the model configuration of YOLOv5 which will be used in this study.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X



```yaml
# Parameters
nc: 4  # number of classes
depth_multiple: 1.33  # model depth multiple
width_multiple: 1.25  # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [
   [-1, 1, MobileNet1, [24]],  # 5-P4/16
   [-1, 1, MobileNet2, [48]],  # 7-P5/32
   [-1, 1, MobileNet3, [576]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]], # 4
   [-1, 1, nn.Upsample, [None, 2, 'nearest']], # 5
   [[-1, 1], 1, Concat, [1]],  # 6 cat backbone P1
   [-1, 3, C3, [512, False]],  # 7

   [-1, 1, Conv, [256, 1, 1]], # 8
   [-1, 1, nn.Upsample, [None, 2, 'nearest']], # 9
   [[-1, 0], 1, Concat, [1]],  # 10 cat backbone P0
   [-1, 3, C3, [256, False]],  # 11 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]], # 12
   [[-1, 7], 1, Concat, [1]],  # 13 cat head P4
   [-1, 3, C3, [512, False]],  # 14 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]], # 15
   [[-1, 3], 1, Concat, [1]],  # 16 cat head P5
   [-1, 3, C3, [1024, False]],  # 17 (P5/32-large)

   [[11, 14, 17], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

Figure 5 YOLOv5 Model Config

Model Training

Before performing the model training process several hyperparameter configurations are set beforehand. The hyperparameters used here will be used as a reference for the comparison of training results before and after backbone changes. The hyperparameters used are learning rate, optimizer, epoch, and batch size configurations, Image size or default from YOLOv5. Hyperparameter configuration details can be viewed in Table 1.

Table 1. Hyperparameter configuration

Parameter	Value
Learning rate	0.01
Optimizer	SGD
Epoch	100
Batch size	16
Image size	640
Iou_t	0.2

Performance Evaluation

After training and testing, the next stage is performance evaluation. At this stage, the model evaluation is visualized using the precision-recall curve. data from the precision-recall curve is then used to calculate the precision (1), recall (2), and MAP or Mean Average Precision (3) of each class.

$$Precision = \frac{TP}{(FP+TP)} \quad (1)$$

$$Recall = \frac{TP}{(FN+TP)} \quad (2)$$

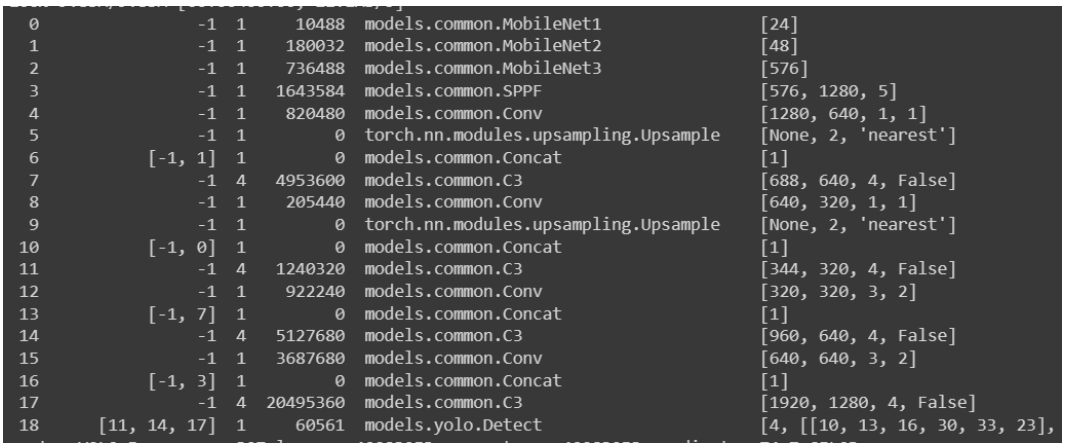
$$MAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (3)$$

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Precision is a measure of the extent to which the model correctly classifies or predicts positives, out of all examples classified as positive. A high precision value indicates that the model tends to have few errors in classifying the wrong attribute as positive. The recall is about how good the model is at finding positive examples. A high recall value indicates that the model tends to have few errors in classifying positive examples as negative.

RESULT

A total of 1624 images were used in the dataset for the experimental results of this study. The dataset was split into 80% for training, 14% for validation and 8% for testing. The architecture of YOLOv5 after backbone change by MobileNet V3 Small is shown in the figure below.



0	-1	1	10488	models.common.MobileNet1	[24]
1	-1	1	180032	models.common.MobileNet2	[48]
2	-1	1	736488	models.common.MobileNet3	[576]
3	-1	1	1643584	models.common.SPPF	[576, 1280, 5]
4	-1	1	820480	models.common.Conv	[1280, 640, 1, 1]
5	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
6	[-1, 1]	1	0	models.common.Concat	[1]
7	-1	4	4953600	models.common.C3	[688, 640, 4, False]
8	-1	1	205440	models.common.Conv	[640, 320, 1, 1]
9	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
10	[-1, 0]	1	0	models.common.Concat	[1]
11	-1	4	1240320	models.common.C3	[344, 320, 4, False]
12	-1	1	922240	models.common.Conv	[320, 320, 3, 2]
13	[-1, 7]	1	0	models.common.Concat	[1]
14	-1	4	5127680	models.common.C3	[960, 640, 4, False]
15	-1	1	3687680	models.common.Conv	[640, 640, 3, 2]
16	[-1, 3]	1	0	models.common.Concat	[1]
17	-1	4	20495360	models.common.C3	[1920, 1280, 4, False]
18	[11, 14, 17]	1	60561	models.yolo.Detect	[4, [[10, 13, 16, 30, 33, 23],

Figure 6 Summary Architecture YOLOv5

By comparing improved YOLOv5 with YOLOv5, we found that YOLOv5 with MobileNet V3 small backbone scored 0.912 at a larger mAP50 than the original YOLOv5 which scored 0.882 at mAP50 with the same training configuration. By comparing the improved YOLOv5 with the original YOLOv5, the result is that YOLOv5 with the small MobileNet V3 backbone gets 0.912 at mAP50 which is greater than the original YOLOv5 which gets 0.882 at mAP50 with the same training configuration. Evaluating and comparing the two types of models trained using the evaluation metrics mAP@50, weight size, memory usage, and training time shows that the model with the MobileNet backbone is better than the original YOLOv5 model. In terms of accuracy on the mAP50 evaluation metric as previously described, the accuracy is 0.912 for YOLOv5 MobileNet and 0.882 for the original YOLOv5. For weight size and memory usage during the training process, YOLOv5 MobileNet itself gets a size of 80.7 MB, for memory usage consumes memory of <7 GB. As for the original YOLOv5 as stated in Table 2, the weight size gets a size of 92.7 MB and memory usage takes up the memory of > 10 GB. See Table 2 for details.

Table 2. Result Improved YOLOv5 and YOLOv5

	YOLO + MobileNet V3 small	YOLOv5
mAP50	0.912	0.882
Weight size	80.7 MB	92.7 MB
Memory usage	6.47 GB	10.9 GB
Training time	1h 11min 19s	49min 36s

First, regarding the results of the original YOLOv5, the overall model achieved a mAP50 value of 0.882 which can also be seen in Figure 7. The model also obtained a Precision value of 0.892, Recall of 0.819. This shows that the model can detect and recognize objects with high accuracy.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

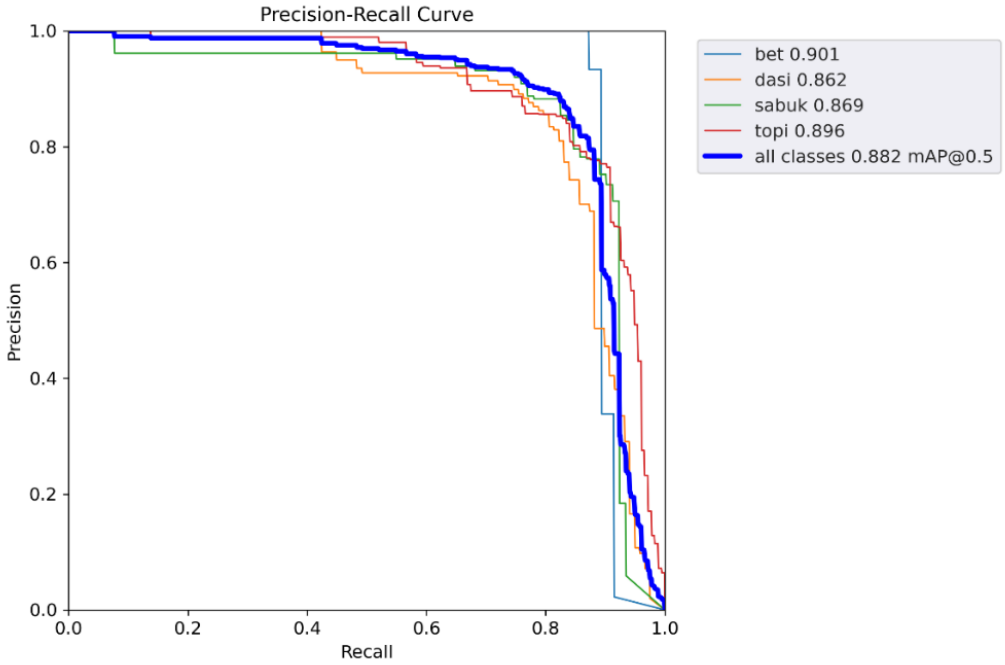


Figure 7. Precision-Recall Curve YOLOv5 Original

Furthermore, the performance evaluation of the system in detecting objects in more specific classes, such as "bet", "dasi", "sabuk", and "topi". As shown in Table 3, the results show that the system has excellent performance in detecting objects in the "bet" class, with Precision (P) values reaching 0.96, Recall (R) of 0.872, and mAP50 of 0.901. This high performance shows that the system can recognize and distinguish objects in the "bet" class very well. However, in the "tie" class, although the system still has good performance with a Precision (P) value of 0.878, Recall (R) of 0.771, and mAP50 of 0.862, there is a slight decrease in object detection performance. The same can be seen in the "belt" class with a Precision (P) value of 0.88, Recall (R) of 0.804, and mAP50 of 0.869. Finally, in class "topi", YOLOv5 model again shows good performance with a Precision (P) value of 0.849, Recall (R) of 0.829, and mAP50 of 0.896. Nevertheless, all classes are still able to provide fairly good results in detecting objects.

Table 3. Details Result from YOLOv5

Class	Precision	Recall	mAP50
All	0.892	0.819	0.882
Bet	0.69	0.872	0.901
Dasi	0.878	0.771	0.862
Sabuk	0.88	0.804	0.869
topi	0.849	0.829	0.896

Furthermore, for the YOLOv5 model with a small MobileNet V3 backbone, the training process is carried out with the same parameters as before. Compared to the previous results, there is an increase in model performance in each class. Overall, as presented in Figure 8, the model achieves high accuracy with a Precision value of 0.914 and a Recall value of 0.873, as well as a mAP50 of 0.912 which is shown in the graphical precision-recall curve quite well. The evaluation results show a considerable improvement in the model's performance in detecting various object classes.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12702
e-ISSN : 2541-2019
p-ISSN : 2541-044X

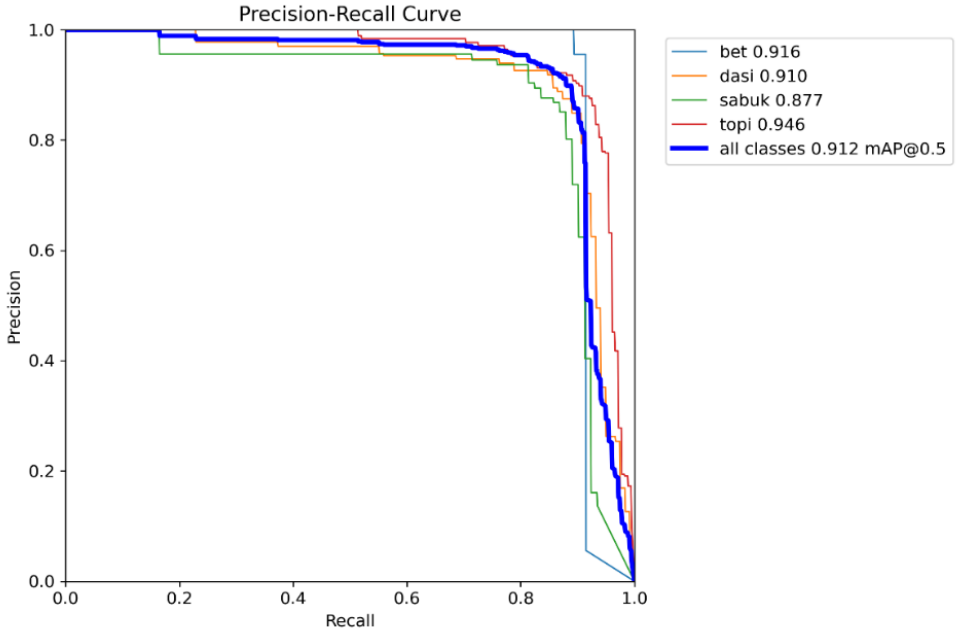


Figure 8 Precision Recall Curve YOLOv5 MobileNet

As can be seen in Table 4, the improved model achieved an overall Precision of 0.914, compared to 0.892 in the previous study. This increase in Precision is also seen in each object class studied, with the highest increase in the "bet" class from 0.96 to 0.979, and the "hat" class from 0.849 to 0.919, which can be seen again in the performance of YOLOv5 in table 3. This improvement shows that the updated model has a better ability to recognize and limit false positives. In addition, the updated model also shows an improvement in Recall on some object classes. The overall Recall increased from 0.819 to 0.873. Significant improvements were also seen in the "tie" class from 0.771 to 0.883, and the "belt" class from 0.804 to 0.835. However, there was a slight decrease in the Recall of the "hat" class from 0.829 to 0.88. This shows that the model can still improve its detection ability on some specific object classes. In addition to the increase in Precision and Recall, mAP50 also experienced a significant increase from 0.882 to 0.912 overall. There is a considerable increase in the "hat" class from 0.896 to 0.946, indicating a better ability to provide an accurate score for object detection.

Table 4. Details Result from YOLOv5 MobileNet

Class	Precision	Recall	mAP50
All	0.914	0.873	0.912
Bet	0.979	0.894	0.916
Dasi	0.874	0.883	0.91
Sabuk	0.885	0.835	0.877
topi	0.919	0.88	0.946

DISCUSSIONS

According to these results, the YOLOv5 model improved with the MobileNet V3 small backbone achieved better performance evaluation results in detecting various object classes in the dataset used compared to the original YOLOv5.. The improvements in Precision, Recall, and mAP50 show significant progress in the model's ability to recognize and detect objects with higher accuracy. YOLOv5 with MobileNet V3 small backbone also shows very satisfactory results, based on Table 2, it can be said that by changing the backbone to MobileNet V3 small the model becomes faster and more efficient than the original YOLOv5. Although the training time is longer than the original YOLOv5, the weight and memory used are much smaller than the original YOLOv5. Although most of the images in the dataset are elementary school uniform images, converting them to grayscale helps the model detect attributes

used in middle and high school uniforms as well. Figure 7 is the result of inferencing attribute detection in high school children.

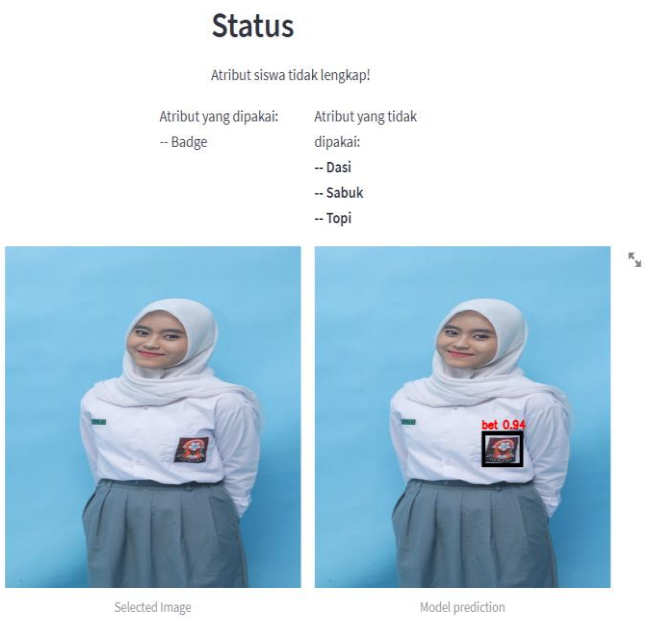


Figure 9 Result of High School Uniform

However, the results of this study are only applicable to the situation and dataset used in this research. Future research will therefore focus on developing more effective and efficient methods to improve model performance, especially for classes of object detection and model sizes that are not yet performance-optimized. Perhaps the use of data augmentation techniques, model architecture optimization, or using of a variety of training data can enhance performance in these classes.

CONCLUSION

In this research, it is shown that YOLOv5 that has been backbone transformed with MobileNet V3 small has higher accuracy results compared to the original YOLOv5. In addition to having high accuracy results, the improved YOLO produces a model that is more efficient in memory usage and the size of its weight. For the accuracy result itself, although the improved YOLOv5 is higher by about 0.912 for mAP50. However, this cannot be generalized to other problems, further research is needed. It can also show that replacing the backbone in YOLOv5 can affect the accuracy of the model.

REFERENCES

Audina, D., Soleh, D. A., & Sumantri, M. S. (2022). Pendidikan Karakter Cinta Tanah Air dan Kedisiplinan Dalam Kegiatan Upacara Bendera di Sekolah Dasar DKI Jakarta. *EduStream: Jurnal Pendidikan Dasar*, *5*(1), 60–68. https://doi.org/10.26740/eds.v5n1.p60-68

Benjumea, A., Teeti, I., Cuzzolin, F., & Bradley, A. (2021). *YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles*. http://arxiv.org/abs/2112.11798

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. http://arxiv.org/abs/2004.10934

Gelar Guntara, R. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *5*(1), 55–60. https://doi.org/10.47233/jteksis.v5i1.750

Hastomo, W., & dan Sudjiran, S. (2021). Convolution Neural Network Arsitektur Mobilenet-V2 Untuk Mendeteksi Tumor Otak. *Seminar Nasional Teknologi Informasi Dan Komunikasi STI&K (SeNTIK)*, *5*(1), 17–21.

Horvat, M., & Gledec, G. (2022). A comparative study of YOLOv5 models performance for image localization and classification. *33rd Central European Conference on Information and Intelligent Systems*, *September*, 349–356. https://github.com/mhorvat/YOLOv5-models-

Howard, A., Wang, W., Chu, G., Chen, L., Chen, B., & Tan, M. (2019). Searching for MobileNetV3 Accuracy vs MADDs vs model size. *International Conference on Computer Vision*, 1314–1324.

Iskandar Mulyana, D., & Rofik, M. A. (2022). Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5. *Jurnal Pendidikan Tambusai*, *6*(3), 13971–13982. https://doi.org/10.31004/jptam.v6i3.4825

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, Hogan, A., Lorenzomammana, Tkianai, YxNONG, AlexWang1900, Diaconu, L., Marc, Wanghaoyang0106, Ml5ah, Doug, Hatovix, Poznanski, J., … Yzchen. (2020). *ultralytics/yolov5: v3.0*. Zenodo. https://doi.org/10.5281/zenodo.3983579

Kim, J. A., Sung, J. Y., & Park, S. H. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. *2020 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2020*, 8–11. https://doi.org/10.1109/ICCE-Asia49877.2020.9277040

Komah, H., & Budjang, G. (2017). *Pengendalian Sosial Oleh Guru Dalam Mengatasi Khulafaur Rasyidin The title of this thesis is social restraint by the teacher in overcoming school attribute infraction in MA Khulafaur Rasyidin. The research problem is how social restraint by the teacher o*. 1–9.

Prayitna, D. H., Djajadi, A., Teknologi, M., Universitas, I., Alamat, P., Business, S., Sangereng, C., Dua, K., & Sekolah, A. (2022). Perancangan Prototype Deteksi Kelengkapan. *Jurnal Inovasi Informatika*, *7*(1), 57–69. https://jurnal.pradita.ac.id/index.php/jii/article/view/235

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. http://arxiv.org/abs/1804.02767

Setiawan, W. (2020). Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus. *Jurnal Simantec*, *7*(2), 48–53. https://doi.org/10.21107/simantec.v7i2.6551

SUYUTI, M. (2023). *Pengembangan Model Klasifikasi Mata Tertutup dan Terbuka Dalam Identifikasi Kelelahan Menggunakan Arsitektur Mobile CNN*. https://dspace.uii.ac.id/handle/123456789/42508%0Ahttps://dspace.uii.ac.id/bitstream/handle/123456789/42508/18917213.pdf?sequence=1&isAllowed=y