Short Communication

Mold Detection on Food Surfaces Using YOLOv5

[a]Md. F. Jubayer[1*], [a]Md. J. A. Soeb[2], Mitun K. Paul[3], Pranta Barua[3], Md. S. Kayshar[1], Md. M. Rahman[2], Md. A. Islam[2]

[1] Department of Food Engineering and Technology; Sylhet Agricultural University, Sylhet-3100, Bangladesh
[2] Department of Farm Power and Machinery, Sylhet Agricultural University, Sylhet-3100, Bangladesh
[3] Department of Electrical and Electronic Engineering, Sylhet Engineering College, Sylhet-3100, Bangladesh
[a] These authors have equal contribution

* Correspondence: Md. F. Jubayer, fahadbau21@hotmail.com; jubayer.fet@sau.ac.bd; Tel.: +8801876529642

Abstract: The study aimed to identify different molds that grow on various food surfaces. As a result, we conducted a case study for the detection of mold on food surfaces based on the "you only look once (YOLO) v5" principle. In this context, a dataset of 2050 food images with mold growing on their surfaces was created. The dataset was trained using the pre-trained YOLOv5 algorithm. In comparison to YOLOv3 and YOLOv4, this current YOLOv5 model had better precision, recall, and average precision (AP), which were 98.10%, 100%, and 99.60%, respectively. The YOLOv5 algorithm was used for the first time in this study to detect mold on food surfaces. In conclusion, the proposed model successfully recognizes any kind of mold present on the food surface. Using YOLOv5, we are currently conducting research to identify the specific species of the detected mold.

Keywords: YOLOv5; object detection; mold; food spoilage; deep learning.

1. Introduction

Mold is dangerous to human health and a serious threat to food supply chains. They can grow on a wide variety of acidic products, including fruits or fruit juices, as well as foodstuffs with intermediate moisture content, like breads and bakery products, which many other microorganisms, such as bacteria, cannot [1]. The prompt detection of fungal presence to prevent further harmful effects is very important in food processing.

In the field of computer vision and image processing, object detection is considered as a significant branch. Object detection algorithms have also received a lot of attention in deep learning [2]. There are roughly two types of visual object detection. The first is the region proposal method (e.g. R-CNN[2], Fast R-CNN[3], Faster R-CNN[4], etc.), and the second is the end-to-end method (e.g. YOLO, SSD, RetinaNet etc.) [3]. In terms of object detection speed, the end-to-end method showed better performance than the region proposal methods [4]. As object detection technology has evolved, the YOLO (You only look once) series of algorithms with very high precision and speed have been used in various scene detection tasks [5]. At the same time, the YOLO system computes all of the image's features and predicts all of the objects. YOLOv5 is the fifth generation of YOLO, written in Python programming language [6]. According to various studies YOLOv5 outperforms the rest of the YOLO model in terms of accuracy and speed [5,6,7]. To the best of our knowledge, this is the first time YOLOv5 has been used to detect mold on food surfaces in the present study. However, in some recent studies, YOLOv5 was used to detect various objects such as apples [5], mushrooms [7], ships [4], face masks [8], safety helmets [9], and vehicles [10], etc. Hence, the aim of this study was to detect the molds from the captured images and count the total number of spores in these images.

2. Experimental Design, Materials and Methods

2.1 Data description

There are two types of data file in the dataset. A) Raw digital images which contain a total of 2050 images with a high diversity of poses, angles, lighting conditions, weather conditions, and backgrounds. The images are all in JPG format. B) Image annotation files containing 2050 images. These files specify the precise locations of the objects with labels in the corresponding image. The annotation was done manually, and the annotated values were also saved in txt files. The data files were divided into two folders: images & labels. Each folder was then divided into two subfolders: train and valid. On the other hand, the dataset preparation consists of four steps: 1) data collection, 2) data preprocessing, 3) data annotation, and 4) data augmentation. The Model building and detection consist of eight steps: 1) importing libraries, 2) importing dataset, 3) cloning YOLOv5 repository, 4) installing required libraries for YOLOv5, 5) training YOLOv5 model with mold dataset, 6) plotting metrics in tensor board, 7) detecting mold in images using trained model, and 8) plotting detected images.

2.2 Importing and installing libraries

Some popular Python libraries were imported for matrix operations, plotting, and file handling.

```
import numpy as np

import matplotlib.pyplot as plt

from glob import glob
```

There is a ‘requirements.txt' file in the YOLOv5 directory that lists the required libraries and versions to run the YOLOv5 model. Required library was installed using the bash command ‘!pip install -qr <directory>’. It is good to mention that ‘pip’ is a python package manager.

```
!pip install -qr '/content/yolov5/requirements.txt'
```

2.2.1 NumPy

NumPy's data formats include matrix and multidimensional arrays. NumPy can perform mathematical operations on arrays such as statistical, algebraic, and trigonometric routines. It provides a highly functional multidimensional array and even the necessary tools for computing and regulating the arrays [11].

2.2.2 Matplotlib

Matplotlib is a portable 2D plotting and imaging package largely used to visualize scientific, engineering, and financial data. The software is free to download, use, and distributeIt has a wide range of applications. Most users are familiar with the command-line interface for creating plots and images interactively. This interface displays and manipulates data in a simple pop-up window [12].

2.2.3 Glob

It finds all path names that match a given pattern.

2.3 Data Collection

Images of foods with mold growing on their surfaces were collected. 750 images of various foods stored in the laboratory of Department of Food Engineering and Technology of Sylhet Agricultural University, Sylhet, Bangladesh were collected. The images were captured using an action camera (GoPro HERO 9 5K Ultra HD). The remaining images were obtained from various internet sources. Both paid and free stock photos were used in this case.

2.4 Data pre-processing and annotation

After taking photos of various mold-affected food surfaces, the images were all converted to JPG format. Tzutalin's [13] popular annotation tool LabelImg was used to cautiously label the images in this final phase. First, each image is opened in this tool one at a time. Then, a rectangular shape was manually drawn to the boundary of an object in order to specify its exact location in that image by x_center y_center width height. Finally, each object has been given a label, such as 'bread mold', or, 'cake mold' etc. In LabelImg, annotated values were saved as txt files in YOLOv5 format.

2.5 Data augmentation

Data Augmentation was done in order to increase the quantity and diversity of data. It aided in the reduction of over fitting in small datasets. To generate new images from the mold dataset, a few data augmentation techniques such as flipping, cropping, and color space transformation were used. The datasets were also uploaded to the roboflow web tool for data augmentation, preprocessing, and train-test split.

2.6 Importing Dataset

The dataset was downloaded to colab using the roboflow generated url as a zip folder. The zip folder has been unzipped and deleted.

```
!curl -L "https://app.roboflow.com/ds/r7gxbFXmxO?key=P4RWGf2c2R" > roboflow.zip;
unzip roboflow.zip; rm roboflow.zip
```

2.7 Cloning YOLOv5 repository

YOLOv5 repository was cloned from GitHub using ‘git clone <url>’.

```
!git clone   'https://github.com/ultralytics/yolov5.git'
```

2.8 Plotting detected images

After running inference, images was saved in ‘runs/detect/exp/’ directory. Using ‘matplotlib’ images were plotted.

```
disp_images = glob('runs/detect/exp/*') # taking location of all file

fig=plt.figure(figsize=(20, 28))

columns = 3

rows = 5

for i in range(1, columns*rows +1):
```

```
img = np.random.choice(disp_images)

img = plt.imread(img)

fig.add_subplot(rows, columns, i)

plt.imshow(img)

plt.show()
```

2.9 Training YOLOv5 model with mold dataset

The mold dataset was designed to train by using Google Colab, which provides free access to powerful GPUs. We used a notebook developed by Roboflow.ai [14] which is based on YOLOv5 [15] and uses pre-trained COCO weights. Suitable number of epochs was chosen to train newly developed mold dataset. To train the model 205 epochs was selected which was taken approximately 40 min. In YOLOv5 directory there is a 'train.py' file for training YOLOv5 model. Using bash command '!python train.py <parameters>' the model was trained accordingly.

The details of training the YOLOv5 model are as follows.

- Image size: 640
- Batch size: 10
- Data description: data.yaml
- Yolo model: YOLOv5s.yaml
- image: height and width of images.
- batch: size of mini batch of images to feed in one iteration.
- epochs: the number of training iteration.
- data: (train, validation) data directory and number of class and class name was described in this YAML file.
- cfg: models are described in YAML model configuration files in 'model' directory. There is four version of model of different size. 'YOLOv5s.yaml' has been used to train.
- name: a model name has been given.

```
%cd yolov5/   # change directory to 'yolov5' using magic command

!python train.py --img 640 --batch 10 --epochs 205 --data /content/data.yaml --cfg models/yolov5s.yaml --name FoodMold
```

2.10 Plotting metrics in tensor board

Tensor Board is a helpful tool for visualizing metrics related to planning, validation, and testing. To visualize mAP, recall, precision metrics 'tensorboard' was loaded using magic command '%load_ext tensorboard'. And '%tensorboard –logdir runs/' was used to visualize metrics from log directory.

```
%load_ext tensorboard
```

doi:10.20944/preprints202105.0679.v1

```
%tensorboard --logdir runs/
```

3. Experimental Outcomes

3.1 Detecting mold in Images using trained model

After training, different types of new and unseen images were used to test the model performance to detect the mold on food surfaces. We took more 100 images as testing set. There is a 'detect.py' file 'yolov5' directory to run inference on images. Inference was performed on the image source directory using '!python detect.py parameters>'.

```
!python detect.py --source /content/valid/images --weights
'runs/train/FoodMold/weights/best.pt
```

Figure 1 shows that the model can detect molds on food surface to a higher degree of certainty.

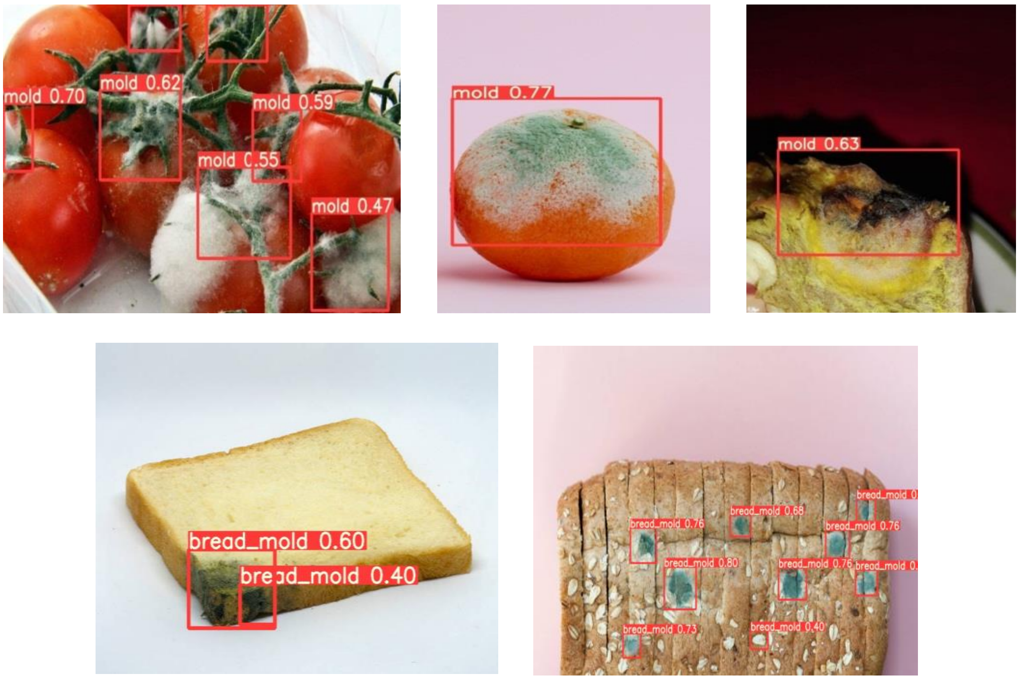

Figure 1. Images from test data sets by showing mold detection on various foods.

3.2 Model performance and comparison

The YOLOv5 model performs really well with 205 epochs. After that with the increase of epochs all the losses like classification loss, box loss and objectness loss were increased. And the model performance was decreased. YOLOv5 was tested to detect mold on various food surfaces after the data set was created. To train the model, various image resolutions were used, and a suitable image resolution of 500×500 pixels was chosen. Before creating the data set, the images were labeled to attain the optimum mold detection. The detected molds' precision, recall, and average precision (AP) were calculated and compared to other models. Results are shown in Table 1. Metrics are calculated with the help of following equations [7].

Precision= TP/(TP+FP)

Recall= TP/(TP+FN)

F1 = [2 / (1/Precision) + (1/Recall)]

AP= 1/11* $\sum r \in (0,0.1,0.2,\ldots..1) p$interp(r)

Table 1. Model performance evaluation with raw dataset under 500×500 pixels' resolution.

Methods	Activation	Precision (%)	Recall (%)	F1 (%)	AP (%)
YOLOv3	Leaky	96.00	95.20	9.50	97.00
YOLOv4	Mish+SPP	98.00	99.05	99.00	99.01
YOLOv5	Mish+SPP	98.10	100.00	99.50	99.60

As a YOLOv5 feature enhancement, the model is better at detecting small to large molds. The activation of Mish+SPP in the models showed an increase in precision, recall, F1-score and AP. It was also discovered that when the image quality was high, the performance was better.

5. Ongoing Research

There are still a lot of things that can be done to improve the mold detection system with YOLOv5. We are currently working on improving the model to detect specific species of food mold. Furthermore, once the system detects and declares any type of food-borne mold, we will attempt to make the entire system more convenient for food industry applications.

Author Contributions: Conceptualization, M.F.J and M.J.A.S; methodology, M.F.J and M.J.A.S.; software, M.K.P and P.B; writing—original draft preparation, M.S.K, M.M.R, and M.A.I; writing—review and editing, M.F.J and M.J.A.S; supervision, M.F.J. All authors have read and agreed to the published version of the manuscript."

Funding: This research received no external funding.

Data Availability Statement: All of the data used to generate the results is included in the article, and no additional source data is required.

Acknowledgments: We gratefully acknowledge the technical assistance provided by the laboratory of Sylhet Agricultural University's Food Engineering and Technology department.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dagnas, S.; Membré, J.M. Predicting and Preventing Mold Spoilage of Food Products. *Journal of Food Protection*. **2013**, 76(3), 538-551. https://doi.org/10.4315/0362-028X.JFP-12-349
2. Pan, C.; Yan, W.Q. Object Detection Based on Saturation of Visual Perception. *Multimedia Tools and Applications*. **2020**, 79(27), 19925-19944. https://doi.org/10.1007/s11042-020-08866-x
3. Wang, L.; Yan, W.Q. Tree Leaves Detection Based on Deep Learning. *Geometry and Vision*. **2021**, 1386, 26. https://doi.org/10.1007/978-3-030-72073-5_3
4. Chen, Y.; Zhang, C.; Qiao, T.; Xiong, J.; Liu, B. Ship detection in optical sensing images based on YOLOv5. In *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)* **2021**, 11720, 117200E. International Society for Optics and Photonics. https://doi.org/10.1117/12.2589395
5. Kuznetsova, A.; Maleva, T.; Soloviev, V. Detecting Apples in Orchards Using YOLOv3 and YOLOv5 in General and Close-Up Images. In *International Symposium on Neural Networks*. **2020**, 233-243. Springer, Cham. https://doi.org/10.1007/978-3-030-64221-1_20
6. Thuan, D. Evolution of YOLO algorithm and YOLOv5: the state-of-the-art object detection algorithm. **2021**.
7. Cengil, E.; Çinar, A. Poisonous Mushroom Detection using YOLOV5. *Turkish Journal of Science and Technology*. **2021**, 16(1), 119-127.

8. Yang, G.; Feng, W.; Jin, J.; Lei, Q.; Li, X.; Gui, G.; Wang, W. Face Mask Recognition System with YOLOV5 Based on Image Recognition. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC).* **2020**, 1398-1404.
9. Zhou, F.; Zhao, H.; Nie, Z. Safety Helmet Detection Based on YOLOv5. In *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA).* **2021**, 22, 6-11. https://doi.org/10.1109/ICPECA51329.2021.9362711
10. Kasper-Eulaers, M.; Hahn, N.; Berger, S.; Sebulonsen, T.; Myrland, Ø.; Kummervold, P.E. Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5. *Algorithms*. **2021**, 14(4), 114. https://doi.org/10.3390/a14040114
11. Sharma, V. Face Mask Detection using YOLOv5 for COVID-19. **2020**. http://hdl.handle.net/20.500.12680/m900nz55w
12. Barrett, P.; Hunter, J.; Miller, J.T.; Hsu, J.C.; Greenfield, P. matplotlib--A Portable Python Plotting Package. In *Astronomical data analysis software and systems*. **2005**, 347, 91-95.
13. Tzutalin. 'LabelImg'. **2018**. Available at: https://github.com/tzutalin/labelImg. (Accessed on November 2020)
14. Roboflow. How to Train YOLOv5 on Custom Objects. **2016**. Available online: https://colab.research.google.com/drive/1gDZ2xcTOgR39tGGs-EZ6i3RTs16wmzZQ (accessed on 3 February 2021)
15. Jocher, G.; Stoken, A.; Borovec, J.; NanoCode012; Christopher, S.; Changyu, L.; et al. Ultralytics/yolov5: v4.0—nn.SiLU() activations, weights & biases logging, PyTorch Hub integration. **2021**. Available online: https://doi.org/10.5281/zenodo.4418161 (accessed on 28 March 2021).