

# CHICKENME: CLASSIFICATION OF CHICKEN DISEASES FROM FECAL IMAGES

DEFENSE EXAM 2023

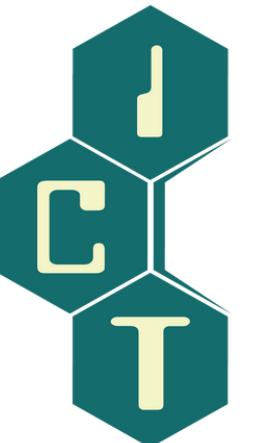
Mr. Waris Damkham 6388014

Miss Pattanan Korkiatrakool 6388022

Mr. Kanokpitch Songdechakaivut 6388058

Advisor: Asst. Prof. Dr. Piyanuch Silapachote

Co - Advisor: Asst. Prof. Dr. Ananta Srisuphab



## CHICKEN ME

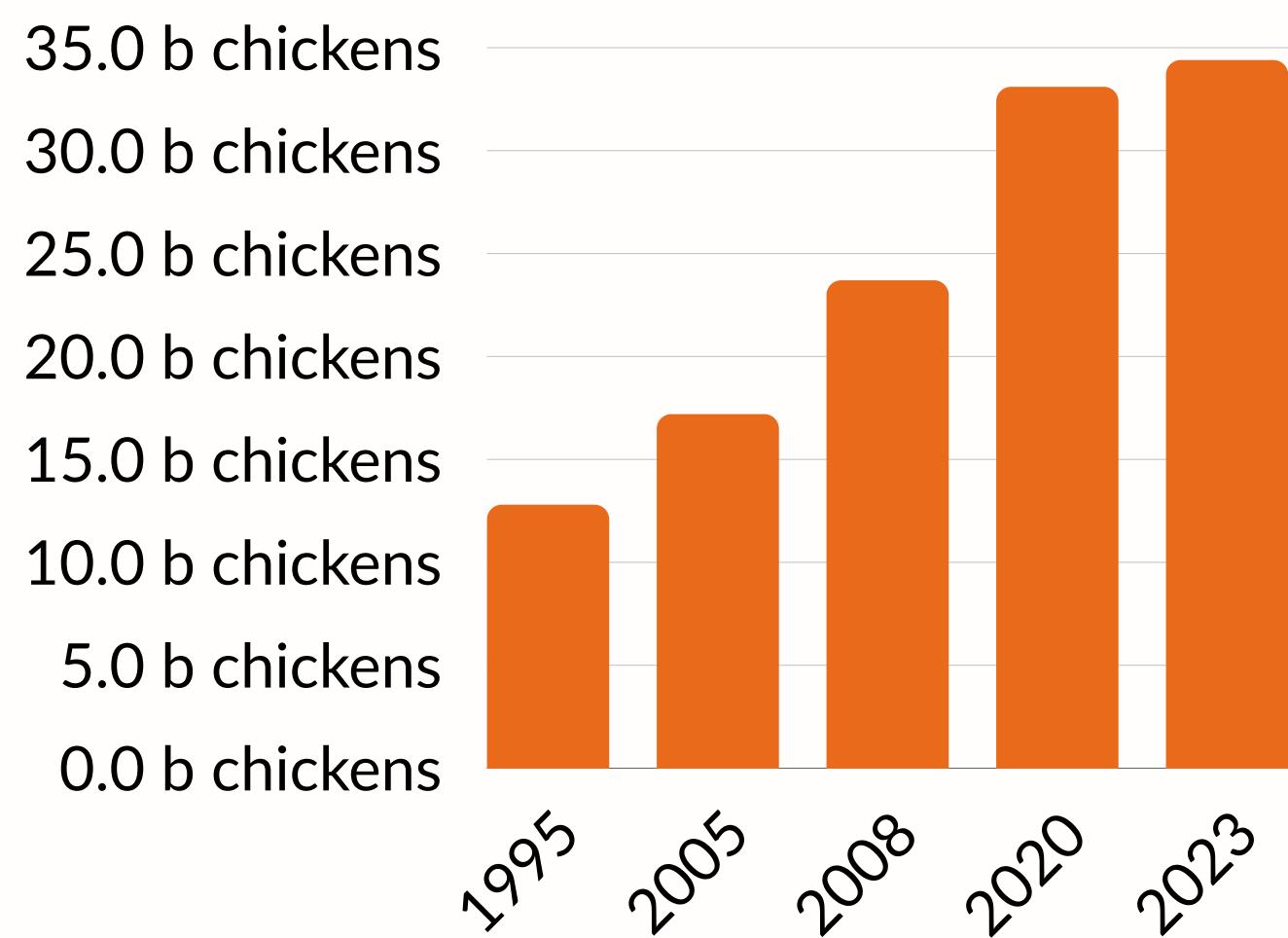
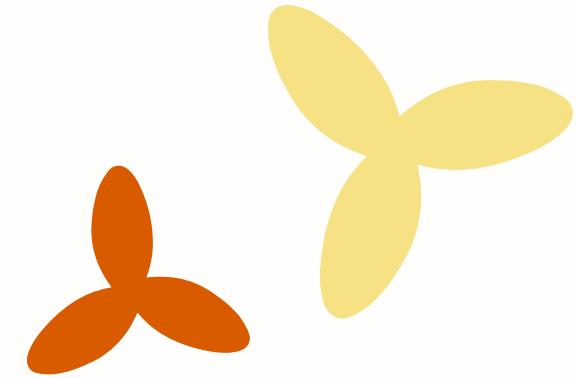
# ABOUT CHICKENME

Smart-farming technology for classification Of  
**Chicken Diseases From Fecal Images**

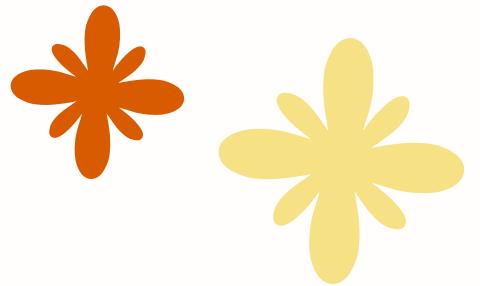
- An automated system for disease classification that detects and predicts health conditions in chickens. The system achieves this by predicting images of chicken droppings, utilized **object detection** and deep learning technology.
- Implemented within the **LINE OA platform** for user-friendly access and real-time communication.



# PROBLEM STATEMENTS



- Approximately 34.4 billion chickens worldwide in 2023
- There are many levels of standards for chicken livestocks
- Some diseases require high epidemic control; **high severity of the disease**
- Disease confirmation using **PCR can be costly**.

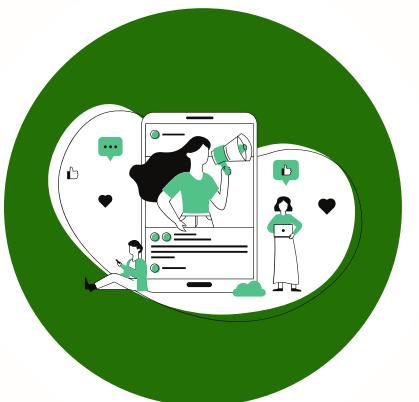


# PROJECT OBJECTIVES



## AUTOMATED DISEASE DETECTION

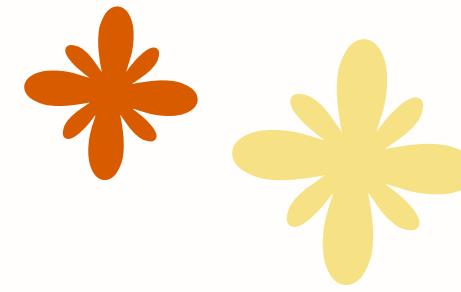
- Utilize Deep Learning to rapidly and accurately segment and detect diseases from chicken-dropping images.



## DELIVER RESULT VIA THE LINE OA

- Implement deep learning within the Line OA
- User-friendly accessibility
- Time-saving experience

# POULTRY DISEASES AND HEALTHY DROPPINGS CHARACTERISTICS



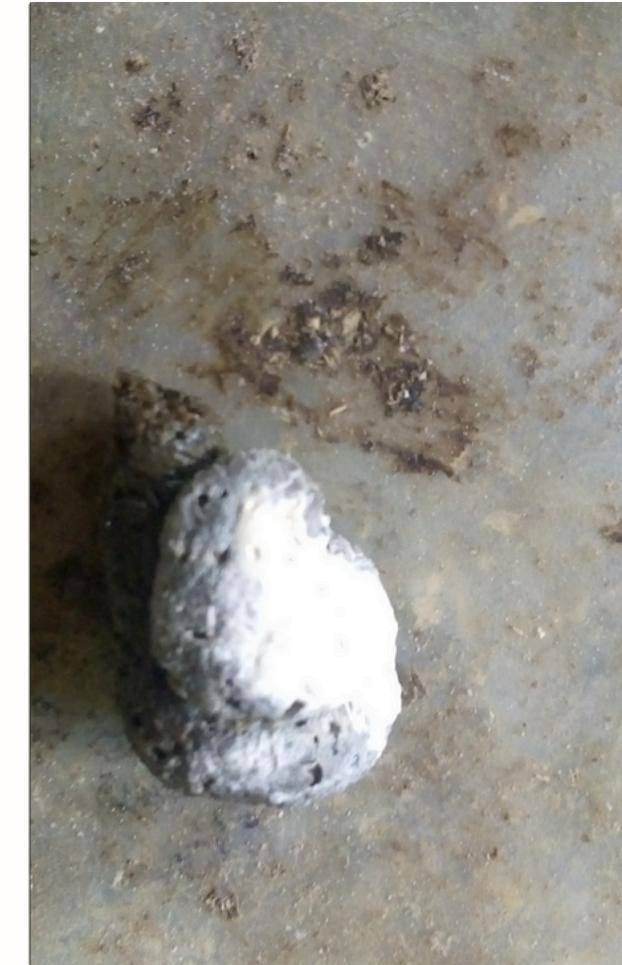
**HEALTHY**

- Well-formed, solid
- various shades of brown.



**COCCIDIOSIS**

- Dark brown, flat feces,
- Diarrhea with blood or mucus
- It is a common disease



**SALMONELLA**

- White, loosely shaped
- It affects people's health and may cause infection

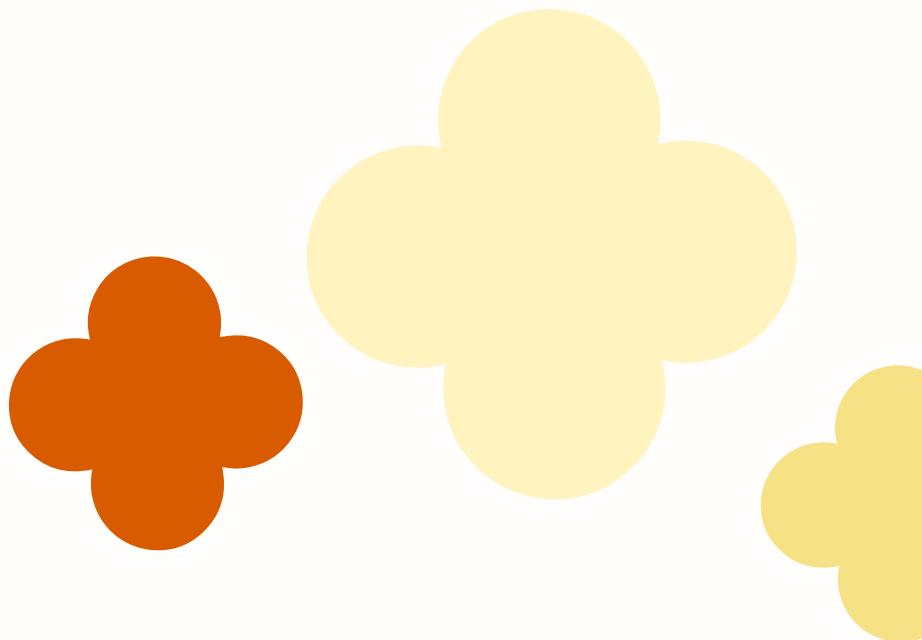


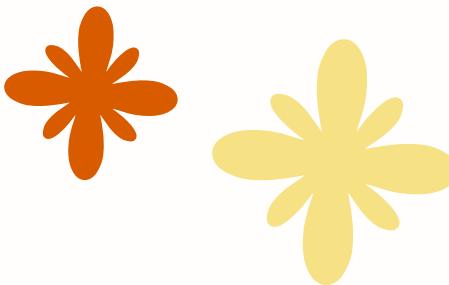
**NEWCASTLE DISEASE  
(NCD)**

- Solid green feces
- The highest level of seriousness can cause death



# RELATED WORK





# RELATED WORK



## DATASET

### Machine Learning Dataset for Poultry Diseases Diagnostics

Machuve, Dina<sup>1</sup> ; Nwankwo, Ezinne<sup>2</sup>; Mduma, Neema<sup>1</sup> ; Mbelwa, Hope<sup>1</sup>; Maguo, Evarist<sup>3</sup>;  
Munisi, Charles<sup>3</sup>

[Hide affiliations](#)

- 1. Nelson Mandela African Institution of Science and Technology
- 2. Duke University
- 3. Elang'ata Agrovet Services

The annotated dataset of poultry disease diagnostics for small to medium-scale poultry farmers consists of poultry fecal images. The poultry fecal images were taken in Arusha and Kilimanjaro regions in Tanzania between September 2020 and February 2021 using Open Data Kit (ODK) app on mobile phones. The typical normal fecal material which is the 'healthy' class and Coccidiosis disease, the 'cocco' class were taken from poultry farms. The chickens were inoculated for Salmonella disease and fecal images taken from the diseased chickens for the 'salmo' class after one week. The chickens were also inoculated for Newcastle disease and fecal images for the 'ncd' class were taken within three days.



## ALGORITHMS



Contents lists available at [ScienceDirect](#)

### Smart Agricultural Technology

journal homepage: [www.journals.elsevier.com/smart-agricultural-technology](http://www.journals.elsevier.com/smart-agricultural-technology)

### Smartphone based detection and classification of poultry diseases from chicken fecal images using deep learning techniques

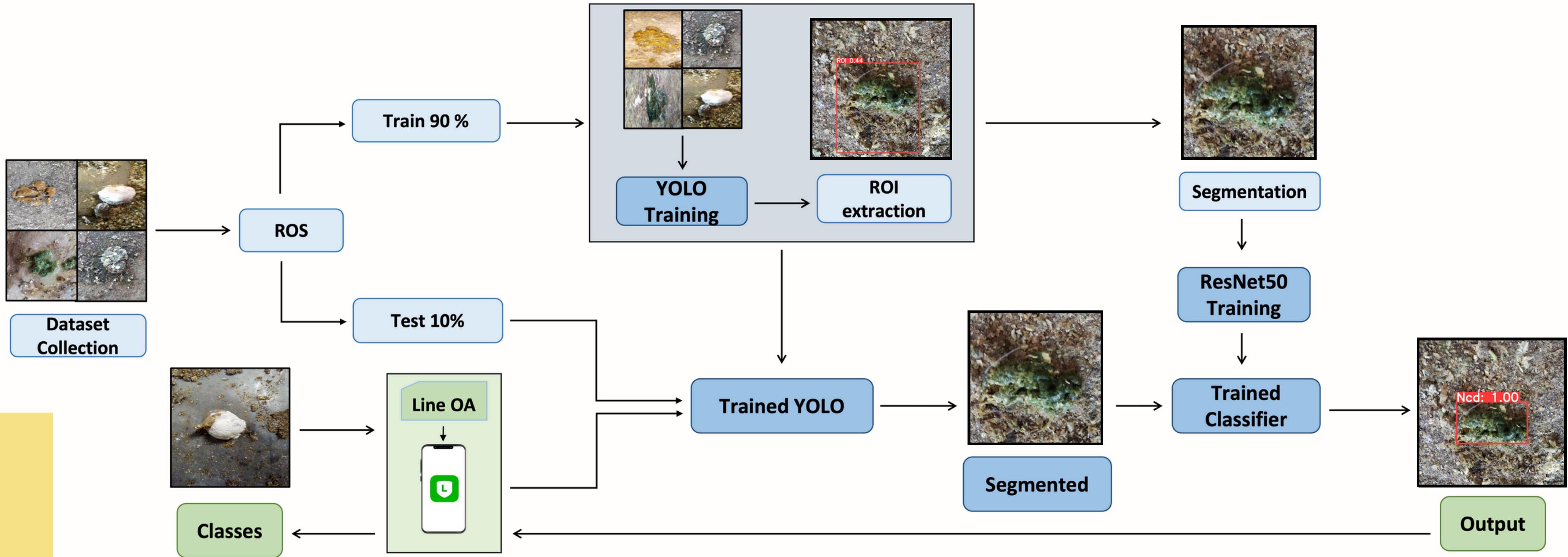
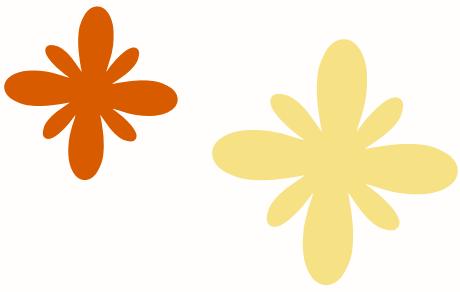
Mizanu Zelalem Degu <sup>a,c,\*</sup>, Gizeaddis Lamesgin Simegn <sup>b,c</sup>

<sup>a</sup> Lecturer of Faculty of Computing and Informatics, Jimma Institute of Technology, Jimma, Ethiopia

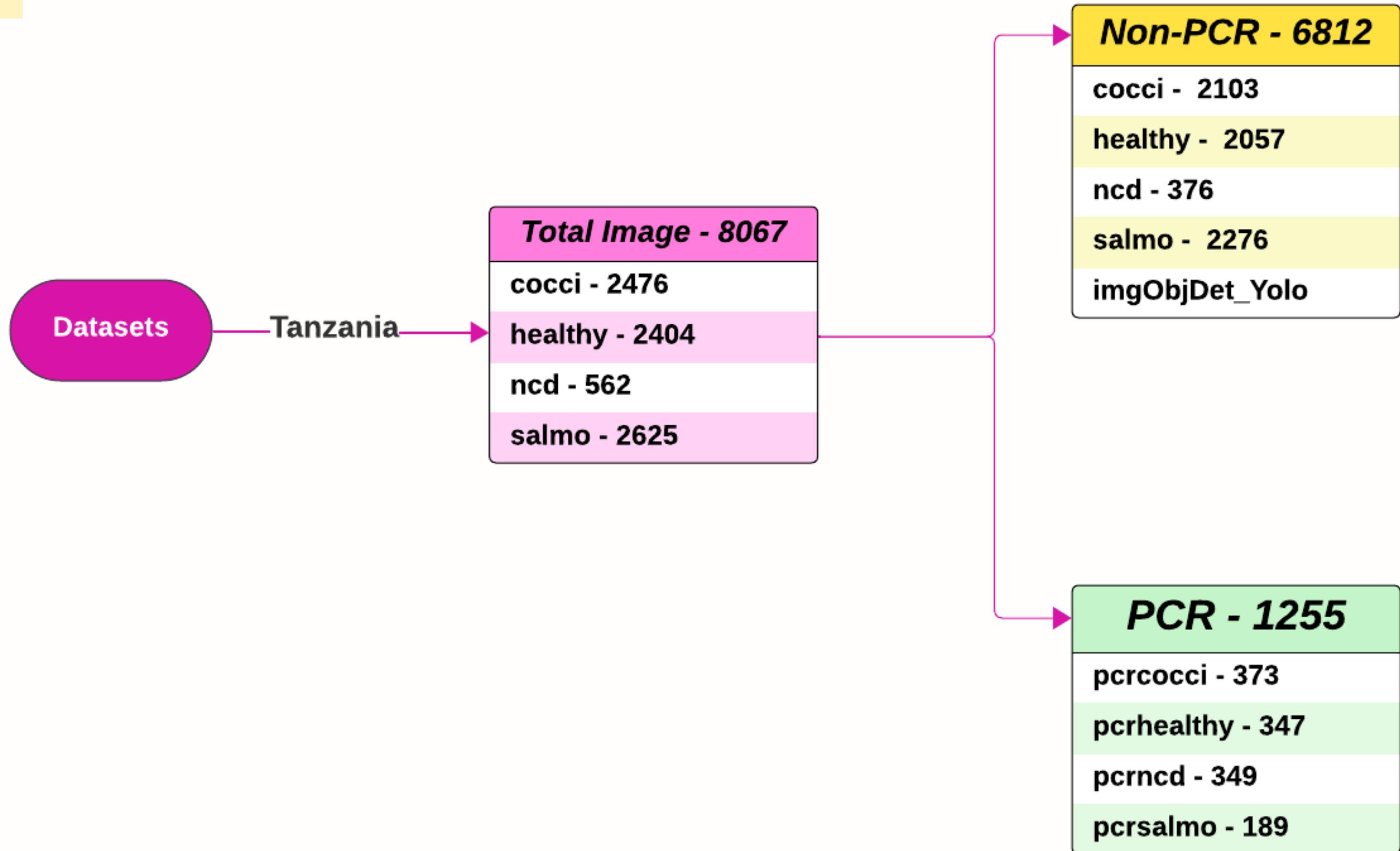
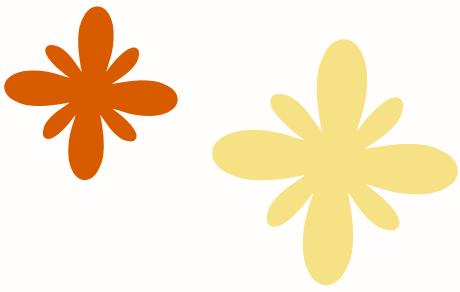
<sup>b</sup> Associate professor of School of Biomedical Engineering, Jimma Institute of Technology, Jimma, Ethiopia

<sup>c</sup> AI and Biomedical Imaging Research Unit, Jimma Institute of Technology, Jimma, Ethiopia

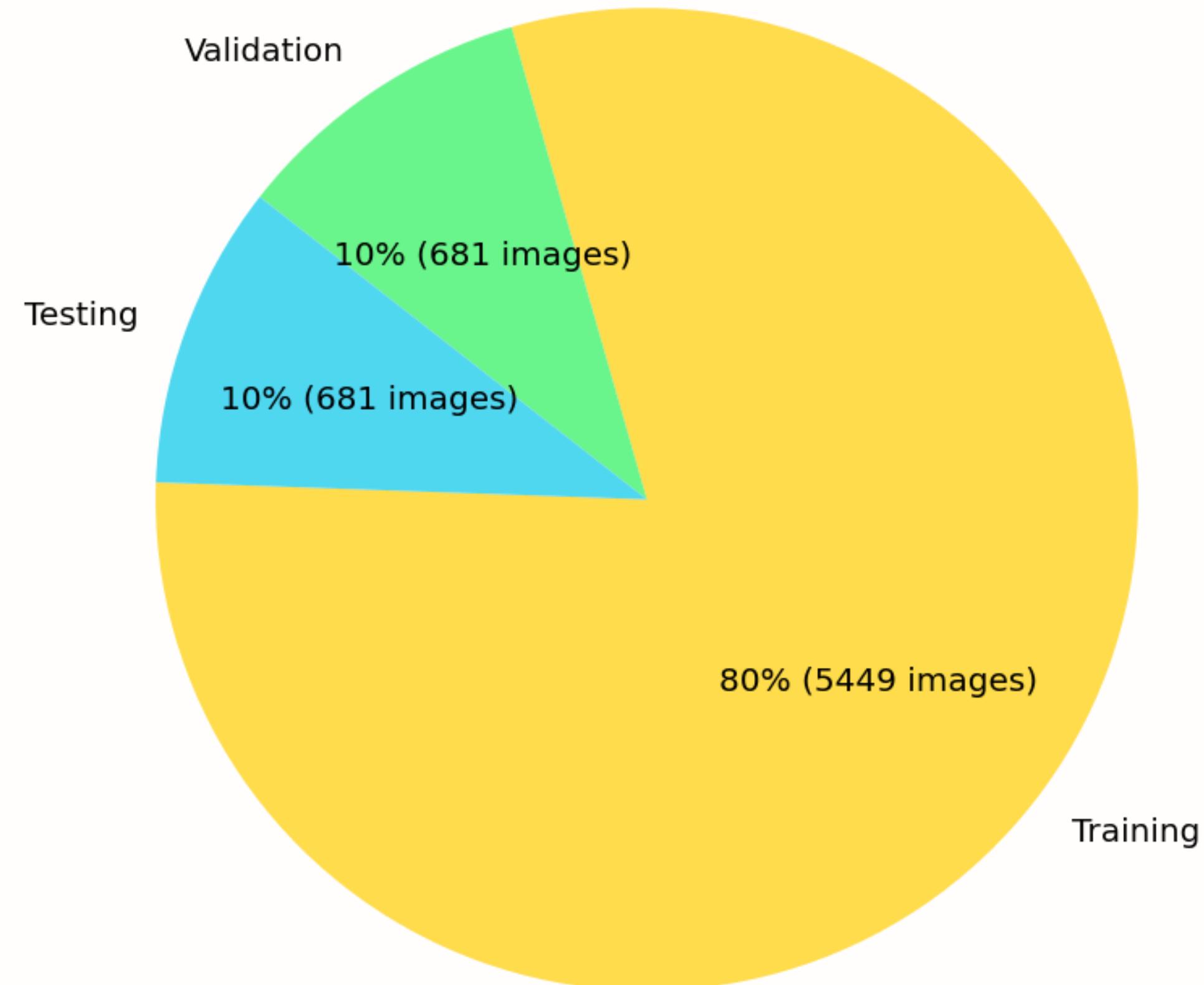
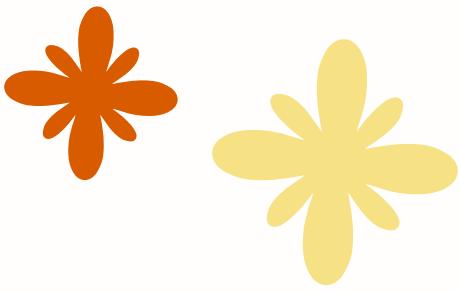
# SYSTEM OVERVIEW



# DATASET

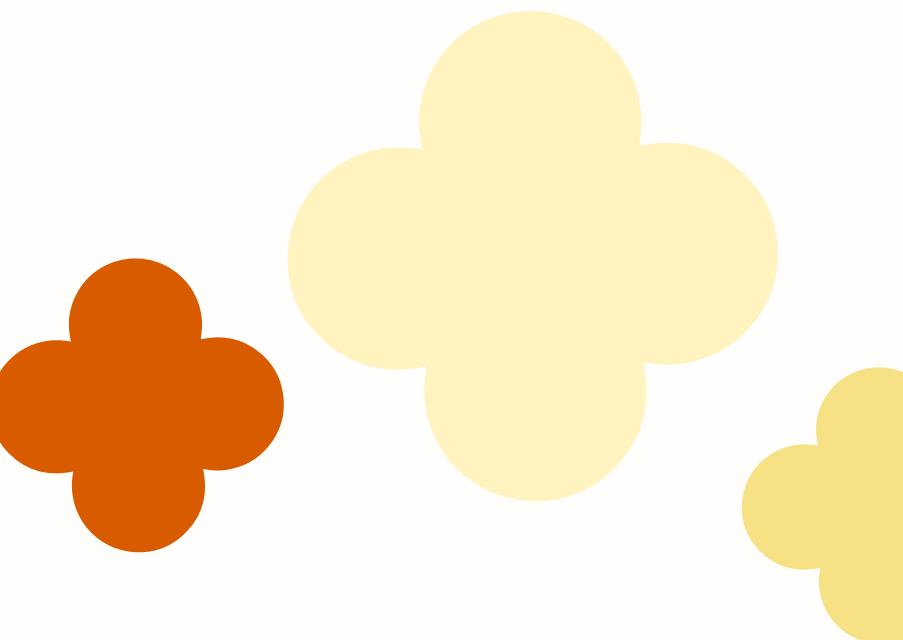
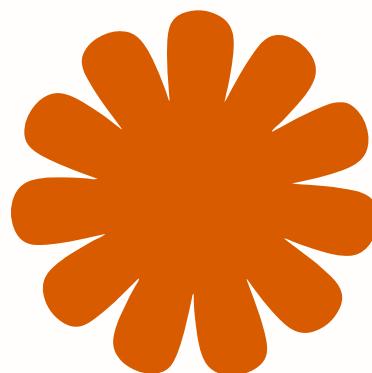


# YOLO DATA SPLITTING

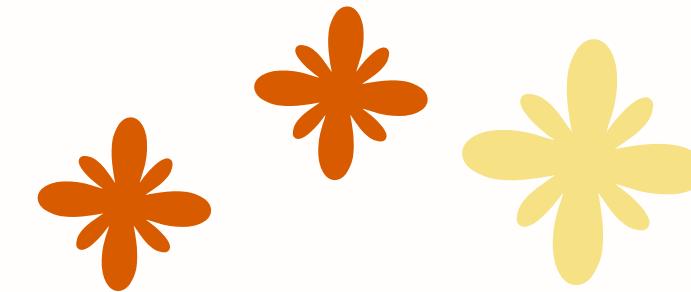




# YOLO MODEL DEVELOPMENT

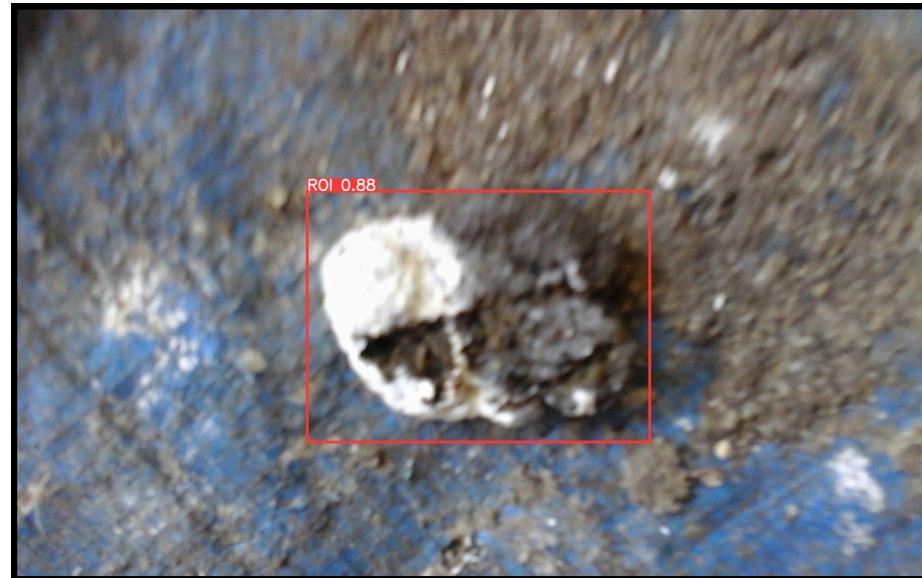


# OBJECT DETECTION / ROI EXTRACTION



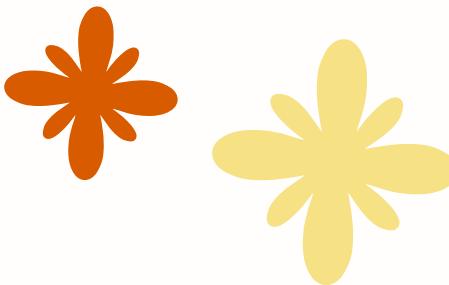
## OBJECT DETECTION

- Detect and locate instances of objects in images or videos.
- Determining the position and boundaries of objects within an image.
- Indicates object positions and provides classification



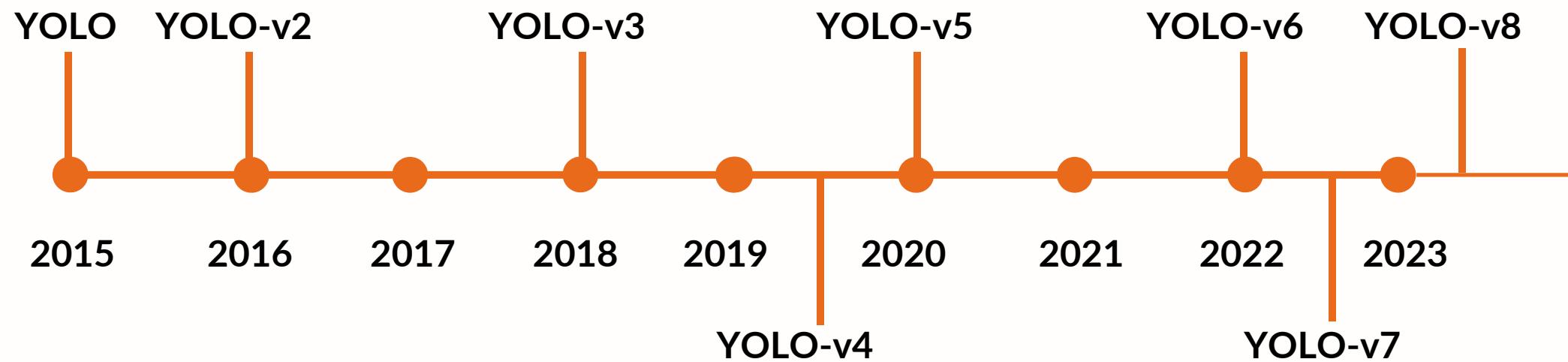
## REGION OF INTEREST (ROI) EXTRACTION

- Identifies and isolates specific areas within an image.
- Avoid the processing of irrelevant image points and accelerate the processing.
- Dividing the image into useful segments.

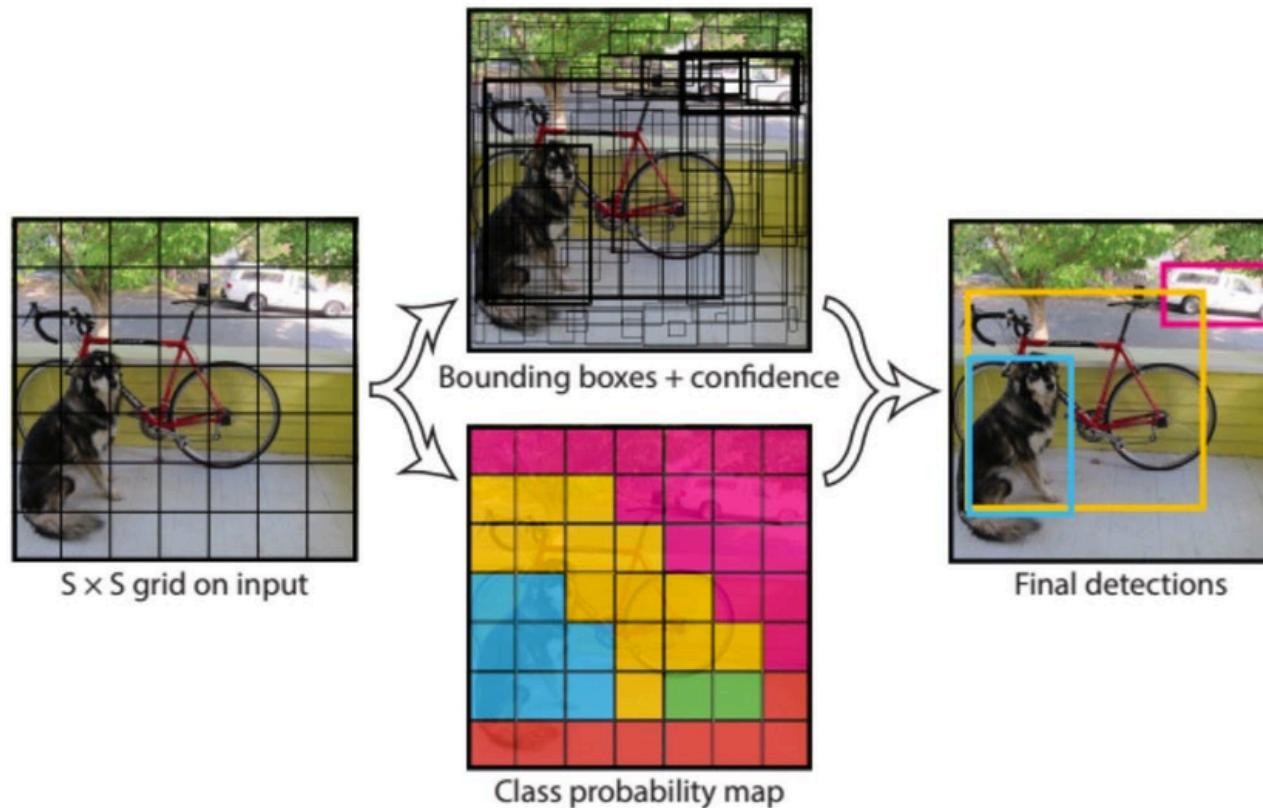


# YOLO (YOU ONLY LOOK ONCE)

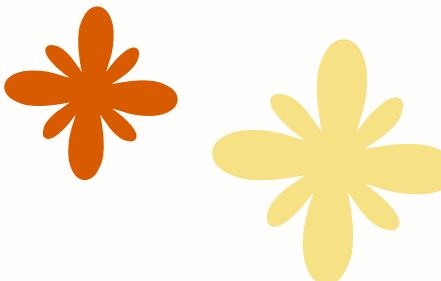
- The high ability to execute real-time object detection tasks.
- Leveraging CNNs for the detection and classification of objects in images.



## YOLO ALGORITHM



- The image is divided into grid cells, each forecasting B bounding boxes with confidence scores.
- Grid cells predict class probabilities to determine the class of each object.



# DARKNET FRAMEWORK AND DARKNET53

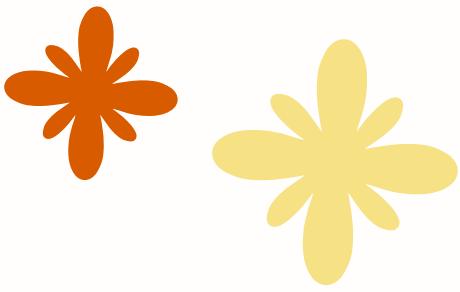
## DARKNET FRAMEWORK

- **Known for:** Association with the YOLO for real-time object detection.
- **Features:** Renowned for its simplicity and efficiency.

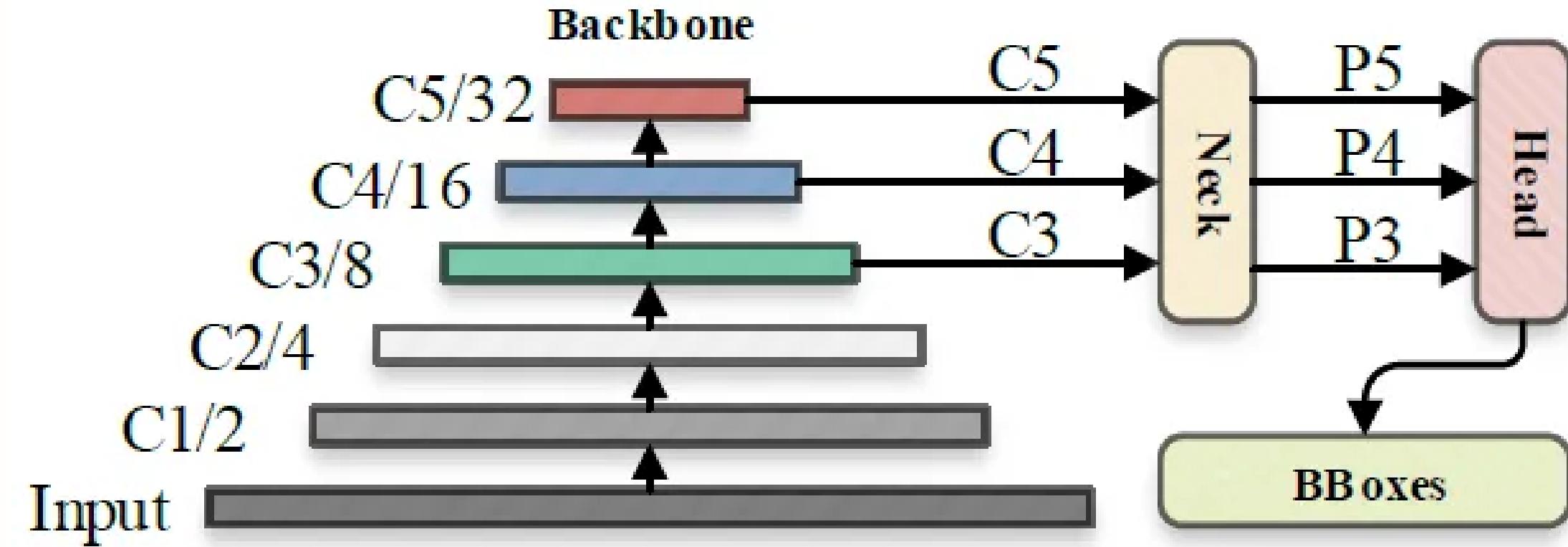
## DARKNET53 ARCHITECTURE

- **Structure:** Consists of 53 layers.
- **Role:** Acts as the backbone or feature extractor in YOLOv3.
- **Attributes:** Lightweight and fast.
- **Usage:** Commonly pre-trained on ImageNet; used as a base for various visual tasks.

	Type	Filters	Size	Output
1x	Convolutional	32	$3 \times 3$	$256 \times 256$
	Convolutional	64	$3 \times 3 / 2$	$128 \times 128$
	Convolutional	32	$1 \times 1$	
	Convolutional	64	$3 \times 3$	
2x	Residual			$128 \times 128$
	Convolutional	128	$3 \times 3 / 2$	$64 \times 64$
	Convolutional	64	$1 \times 1$	
	Convolutional	128	$3 \times 3$	
8x	Residual			$64 \times 64$
	Convolutional	256	$3 \times 3 / 2$	$32 \times 32$
	Convolutional	128	$1 \times 1$	
	Convolutional	256	$3 \times 3$	
8x	Residual			$32 \times 32$
	Convolutional	512	$3 \times 3 / 2$	$16 \times 16$
	Convolutional	256	$1 \times 1$	
	Convolutional	512	$3 \times 3$	
4x	Residual			$16 \times 16$
	Convolutional	1024	$3 \times 3 / 2$	$8 \times 8$
	Convolutional	512	$1 \times 1$	
	Convolutional	1024	$3 \times 3$	
	Residual			$8 \times 8$
	Avgpool		Global	
	Connected			1000
	Softmax			

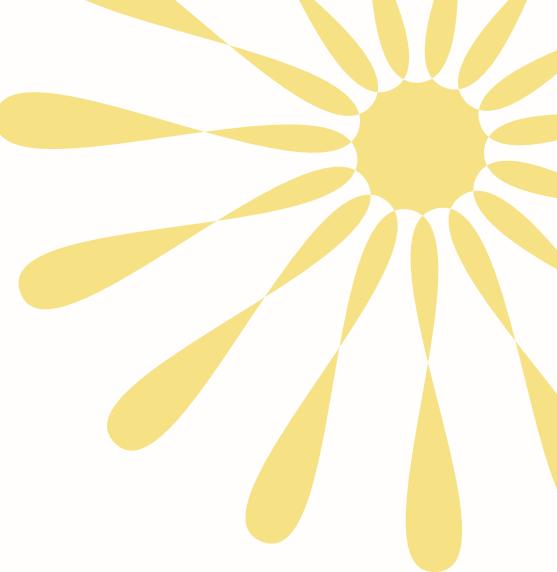


# YOLO V5



- **Backbone:** Extracts features and creates lower-dimensional representations.
- **Neck:** Generate three feature maps P3, P4, and P5 (in the YOLOv5, the dimensions are expressed with the size of 80×80, 40×40, and 20×20) for detecting different scale targets.
- **Head:** Generates bounding boxes, object confidence scores, and class probabilities.

# THE DIFFERENCE BETWEEN YOLOV3 AND YOLOV5



## EASE OF USE

### YOLOv3

- Requires more manual setup
- less streamlined tooling than V5

### YOLOv5

- Easier to use, a simpler development environment.
- Designed with PyTorch, making it more accessible to a broader audience.
- Community support

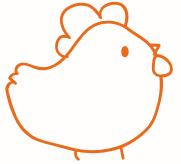
## MODEL SIZE

### YOLOv3

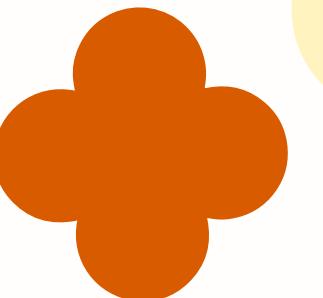
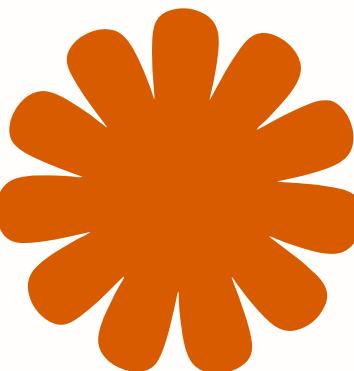
- Larger and heavier
- Resource requirements due to its architecture
  - Complicate deployment

### YOLOv5

- Lighter and more efficient
- various model sizes from small to extra-large
- easier to deploy on hardware with varying resource constraints.



# YOLO DATA PREPARATION AND PRE-PROCESSING

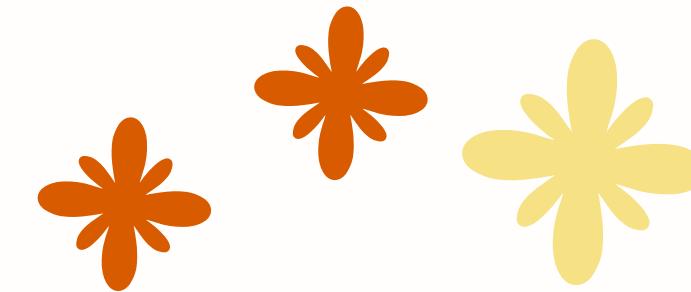


SP2023-31

# BOUNDING BOX



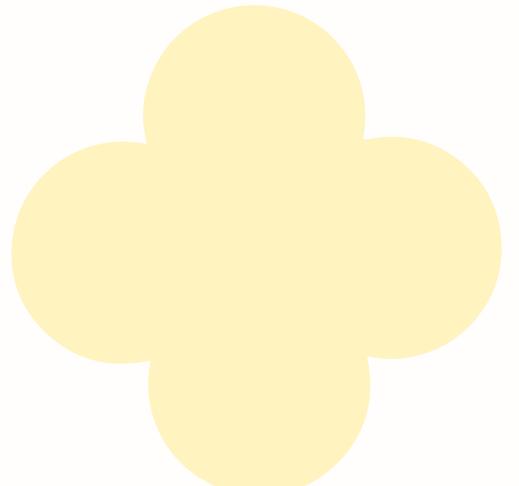
# ROS (RANDOM OVERSAMPLING)



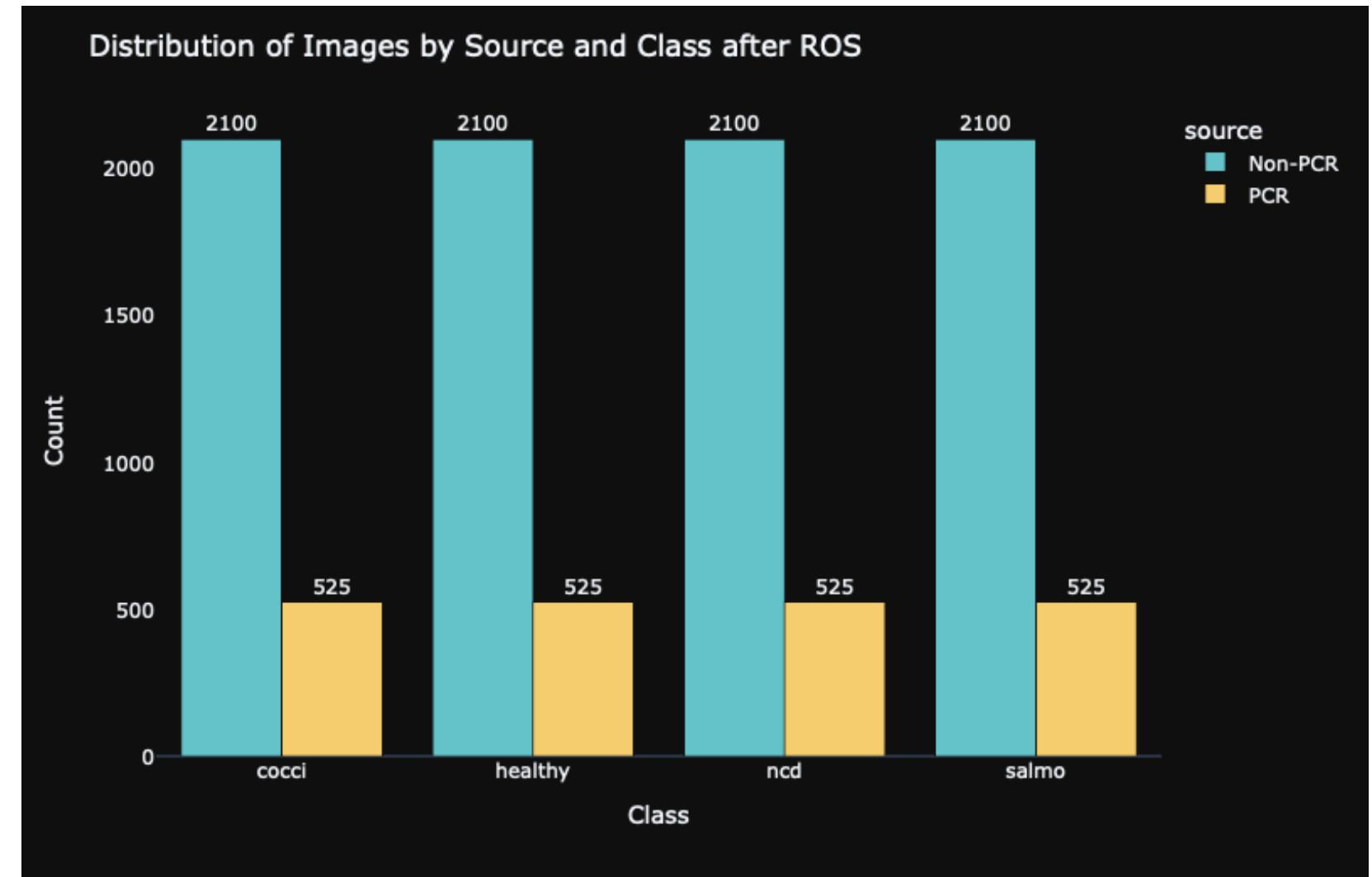
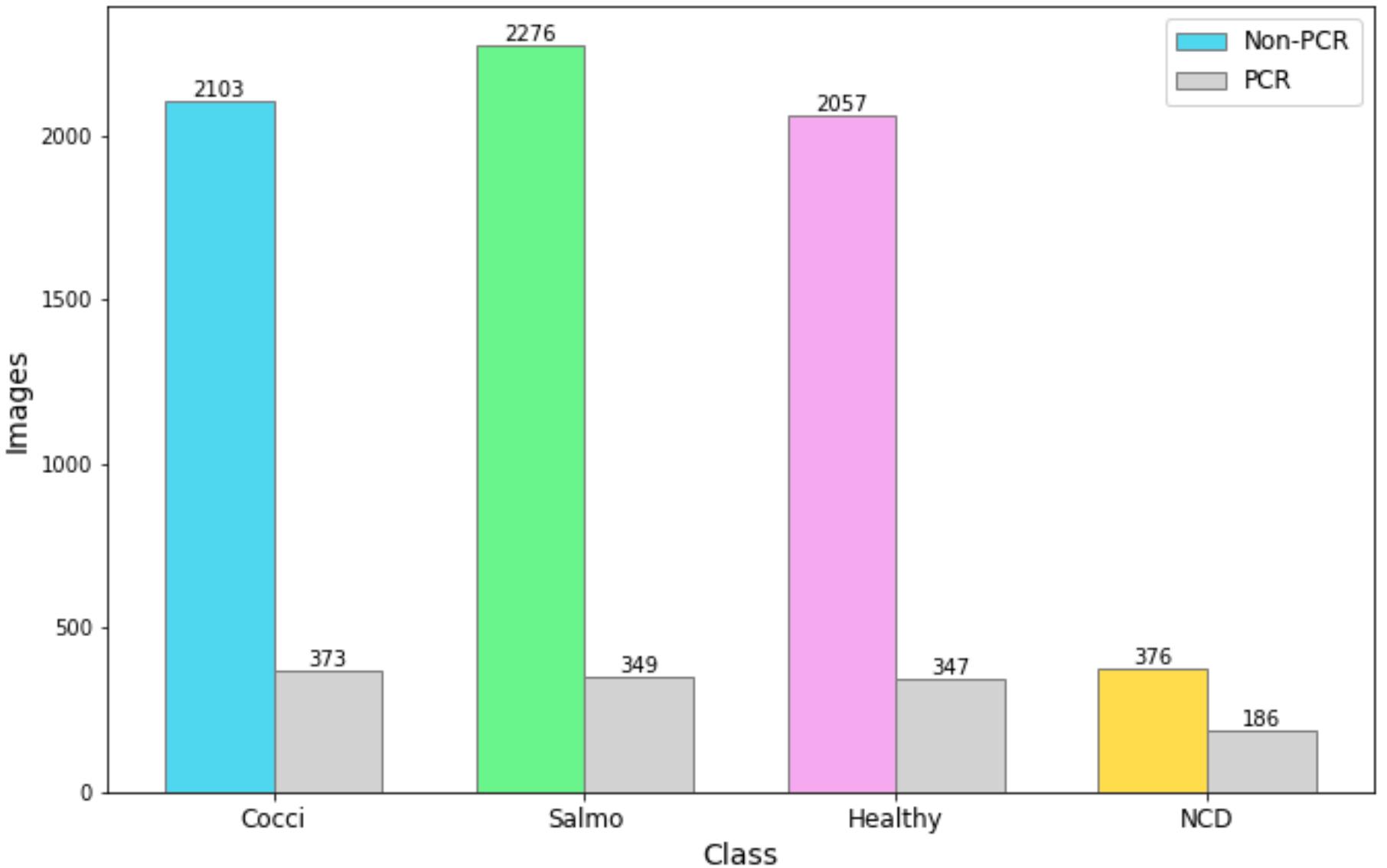
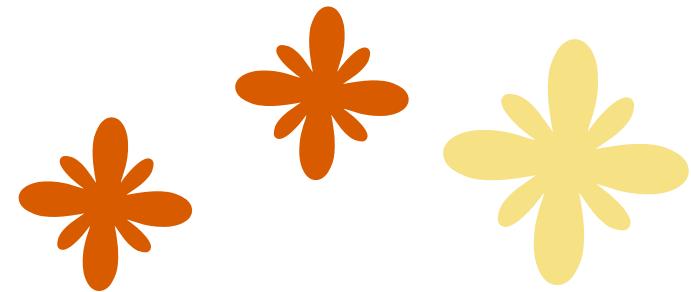
- A technique used to address class imbalance in datasets
- **How it works:** To make the number of samples in the minority class closer to or equal to the number of samples in the majority class.

## BENEFITS:

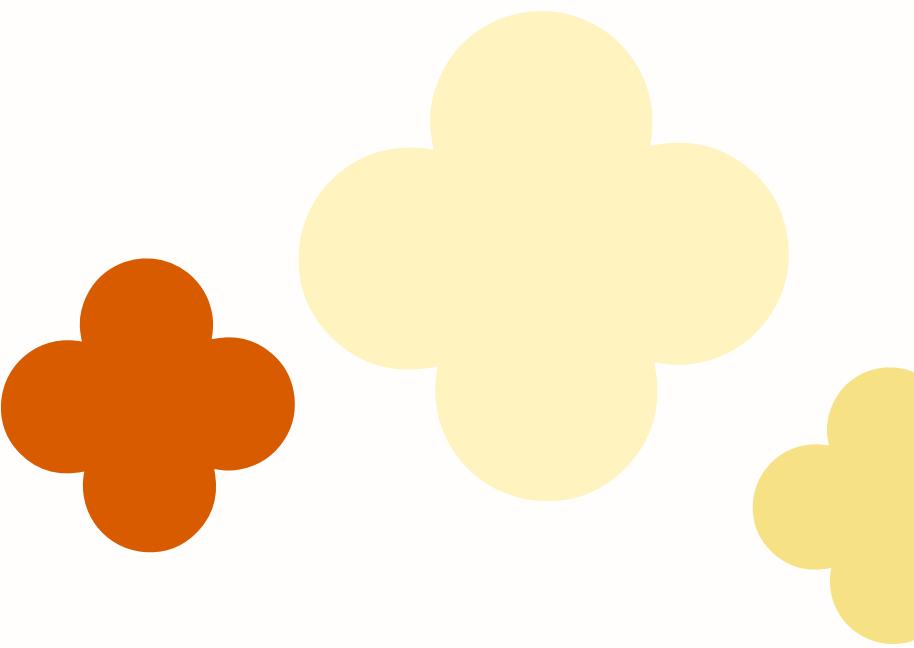
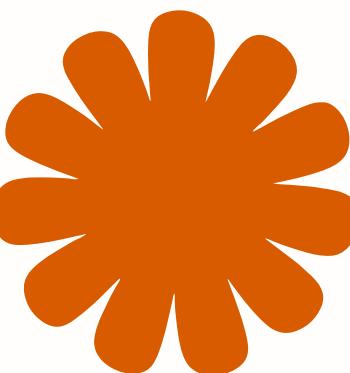
- **Improved Model Training:** Helps models learn features from both minority and majority classes, reducing bias.
- **Reduced Overfitting:** Proper use of ROS oversampling can increase data diversity, mitigating overfitting risks.



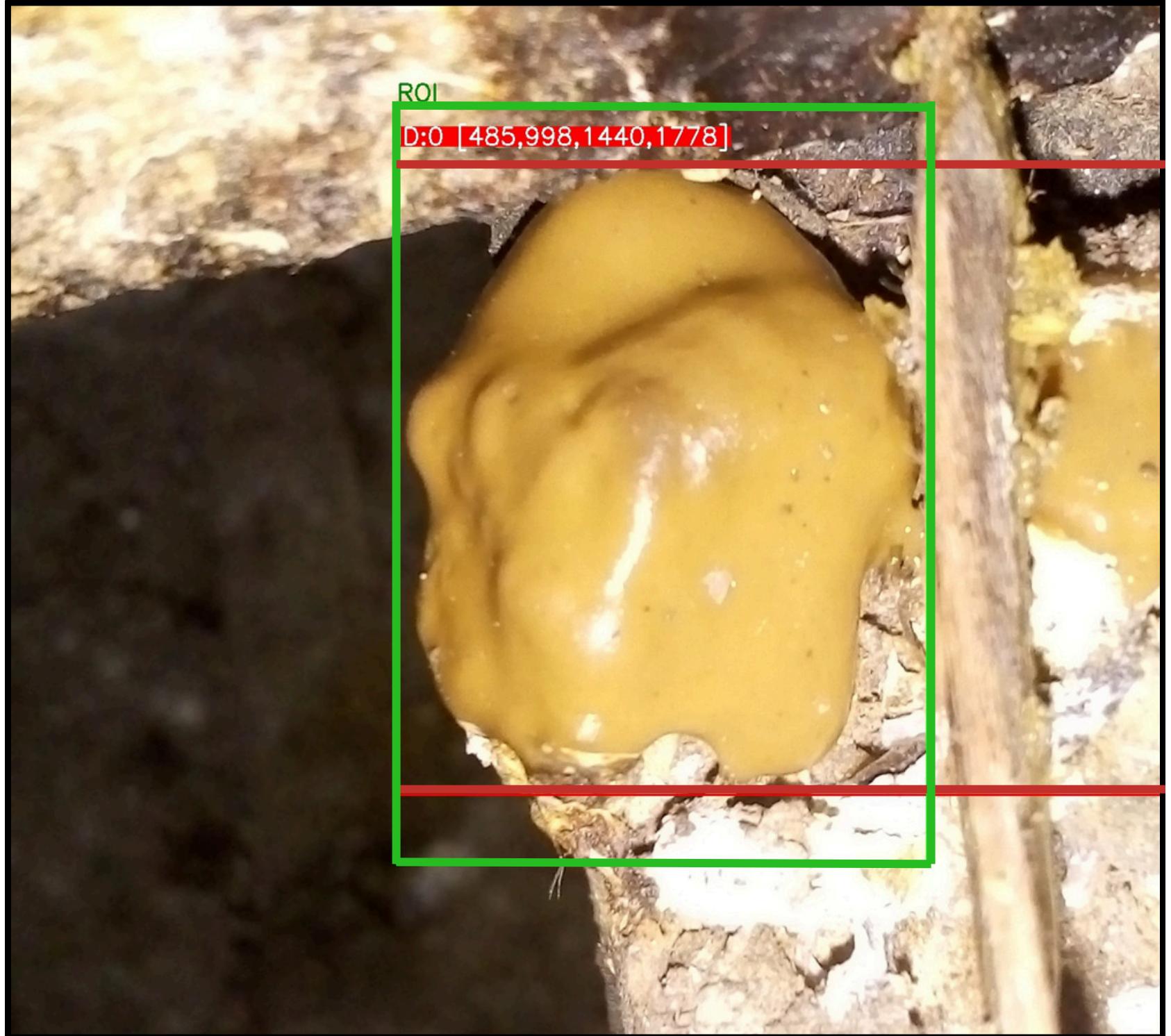
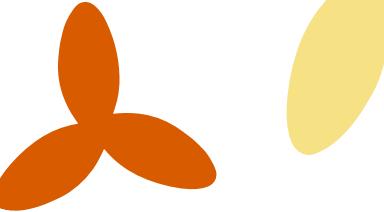
# ROS (CONT.)



# **YOLO** **TRAINING AND** **TESTING PROCESS**



# YOLO MODEL OUTPUT



# MODEL EVALUATION



## MEAN AVERAGE PRECISION (mAP)

- Metric for evaluating object detection models.
- Indicates how well the model detects objects at various confidence levels.



### • INTERSECTION OVER UNION (IOU)

- Assesses the overlap between predicted and ground truth bounding boxes.
  - 1 = perfect overlap
  - 0 = no overlap.

### Calculate Detection Scores

- Determine detection scores for each image in the dataset.

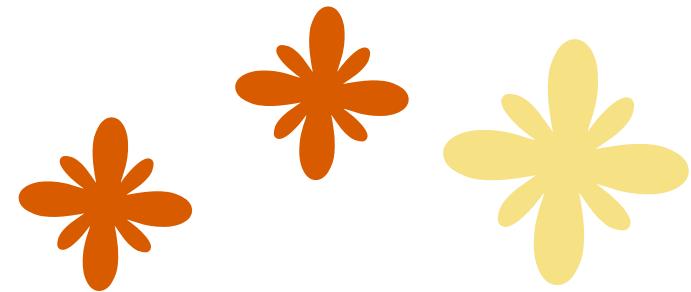
### Compute Error Scores

- $1 - \text{detection score} = \text{error score}$  for each image.

### Compute the Mean and Standard Deviation of IOU

- average (mean) and standard deviation

# YOLOV3 AND YOLOV5 EXPERIMENTS



<b>Dataset</b>	<b>Architecture</b>	<b>Performance on Testing Data</b>	
		<b>Mean Average Precision (mAP)</b>	
456 YOLO-labelled images with full size.	YoloV3	75.31 %	
	Yolov5		<b>82.06 %</b>
6812 YOLO-labelled images with full size.	YoloV3	82.13 %	
	Yolov5		<b>87.10 %</b>

## Training Data

- Both were trained on a dataset of 6812 labeled full-size images from Tanzania, with identical hyper-parameters.

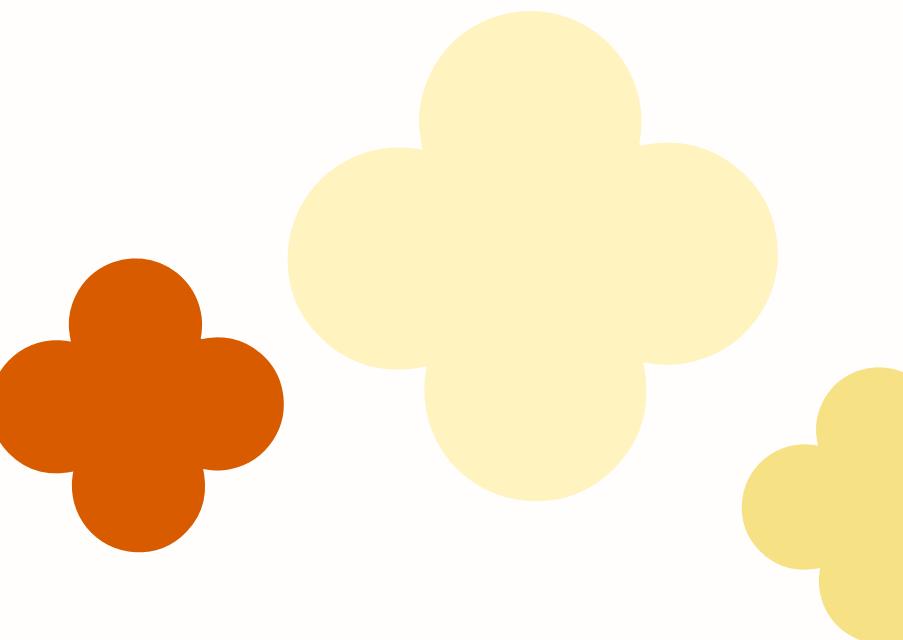
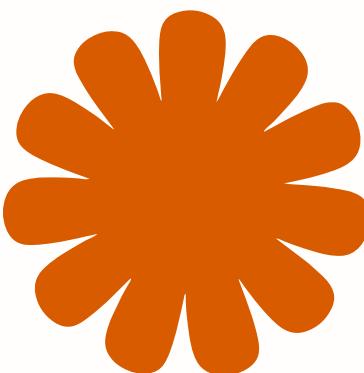
## Testing Data

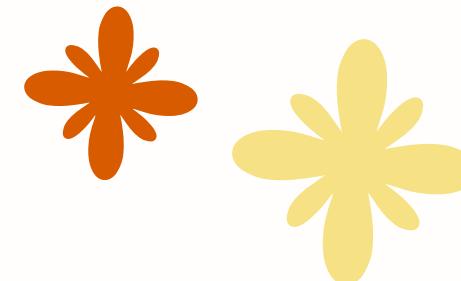
- Tested with 682 unlabeled images, with identical settings and hyper-parameters.

# **RESNET50**

## **MODEL**

## **DEVELOPMENT**



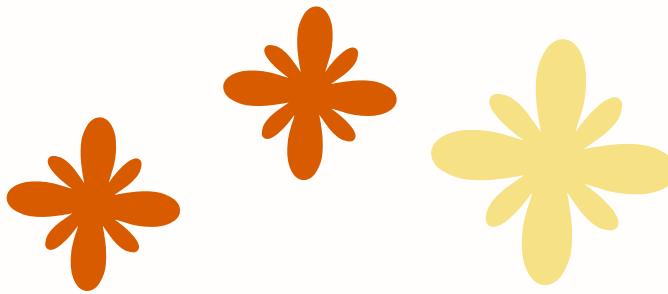


# RESNET50 MODEL : SEGMENTATION FROM YOLOV5



segmented YOLO for RESNET training

- YOLOv5 segments ROI from fecal images after successful training.
- Further processing.
- To extract relevant features from segmented ROI images.
- Learning hierarchical representations of spatial and textural image features.
- uses 10,500 segmented images from YOLOv5, resized to meet its input specifications.



# IMAGE CLASSIFICATION



## CONCEPT

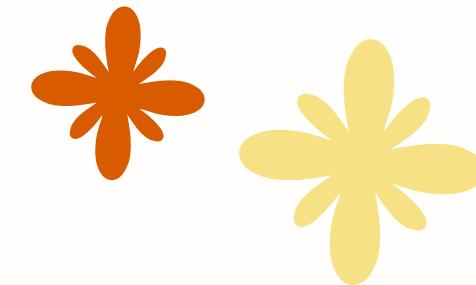
- involves classify an image into predefined classes or labels.
- Categorize an input image by assigning it a specific label or class based on its content.



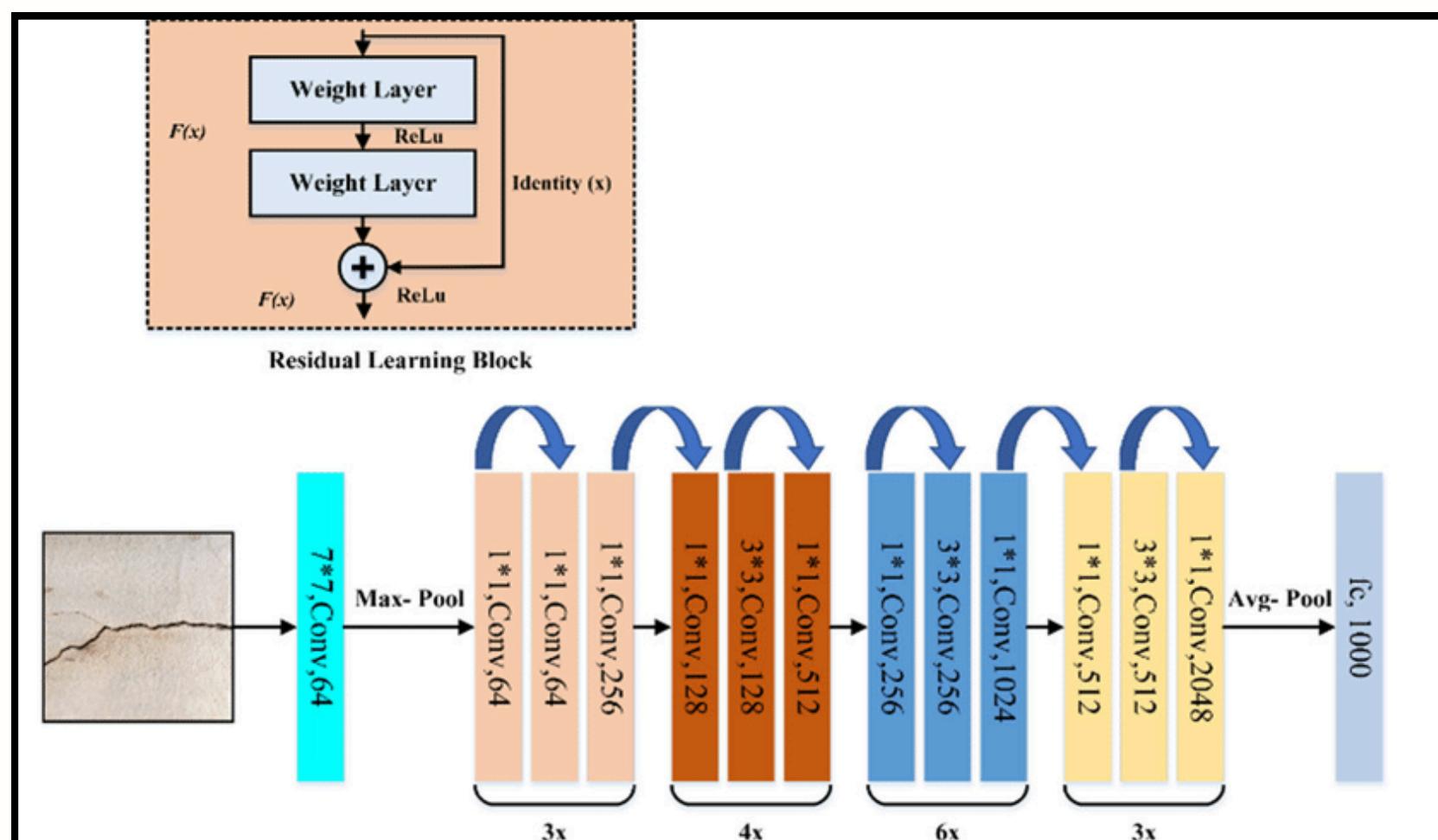
## PROCESS

- Training a Model on a labeled dataset.
- Learns patterns and features that relate to each class.





# RESNET50 MODEL FOR IMAGE CLASSIFICATION



## OVERVIEW OF RESNET50

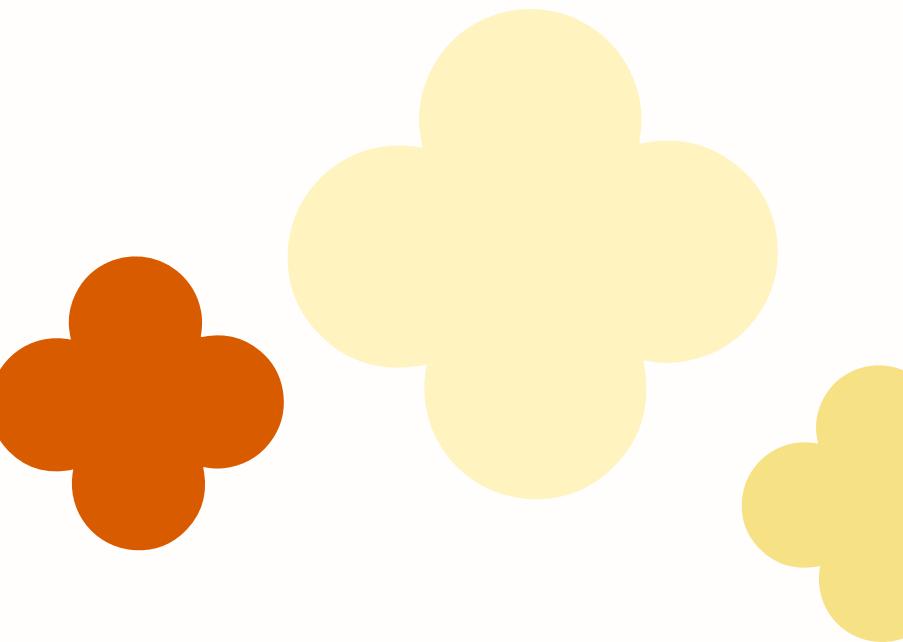
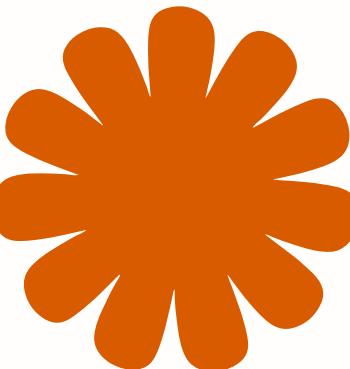
- A deep convolutional neural network with 50 layers.
- Designed for efficient training of deep networks.
- Widely used in various domains including facial recognition and autonomous vehicles.

## KEY FEATURES OF RESNET50

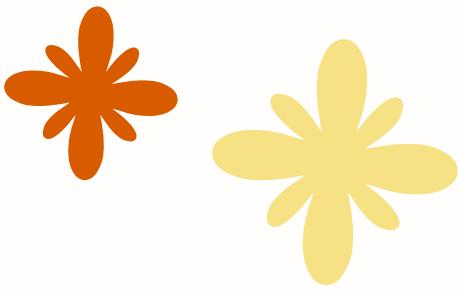
- **Architecture:** 5 stages with convolutional and identity blocks, batch normalization, and ReLU activations.
- **Skip Connections:** Help bypass layers during forward pass to alleviate the vanishing gradient problem.
- **Residual Learning:** Focuses on learning the residual differences between input and output.

# **RESNET50**

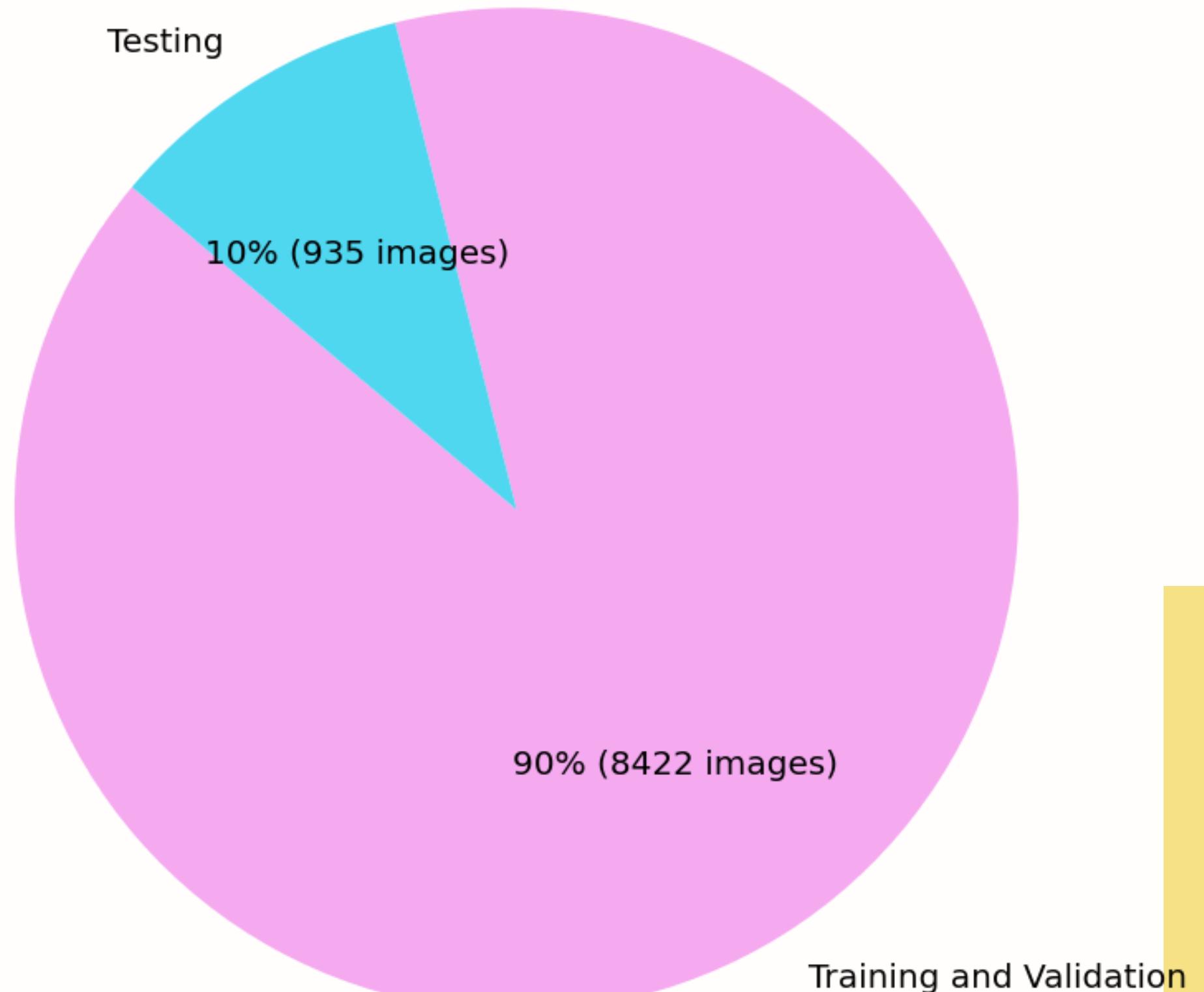
## **TRAINING AND TESTING PROCESS**



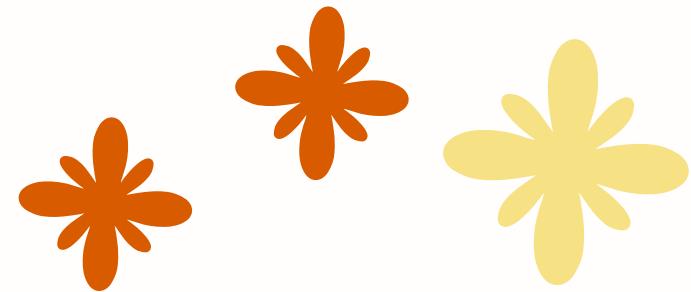
# RESNET DATA SPLITTING



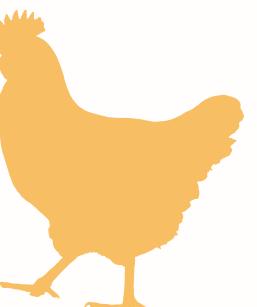
- 9385 segmented images as 90% for **Final Training and Validation**
- 936 segmented images as 10% for **Final Model Testing**
- Validation Data for 10-fold cross-validation
  - 90% is **8422 images**
  - 10% is **842 images** for **Validating in each fold**



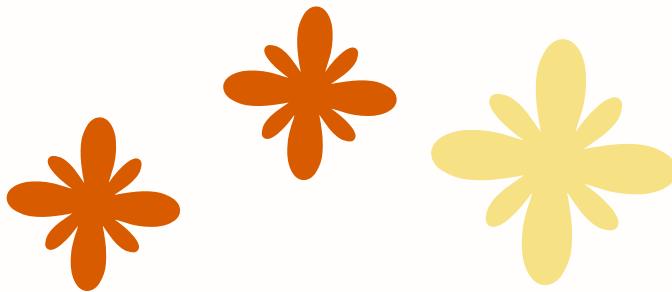
# 10-FOLD CROSS-VALIDATION



- A technique to evaluate model performance
- Dividing the dataset into ten equally sized subsets, known as '**folds**'
- **Trained and evaluated ten times**, with a different fold used as the validation set in each iteration.
- The remaining **nine folds** are used as **training data** for each iteration.
- Aggregated performance metrics from each fold assess the model's generalization ability.



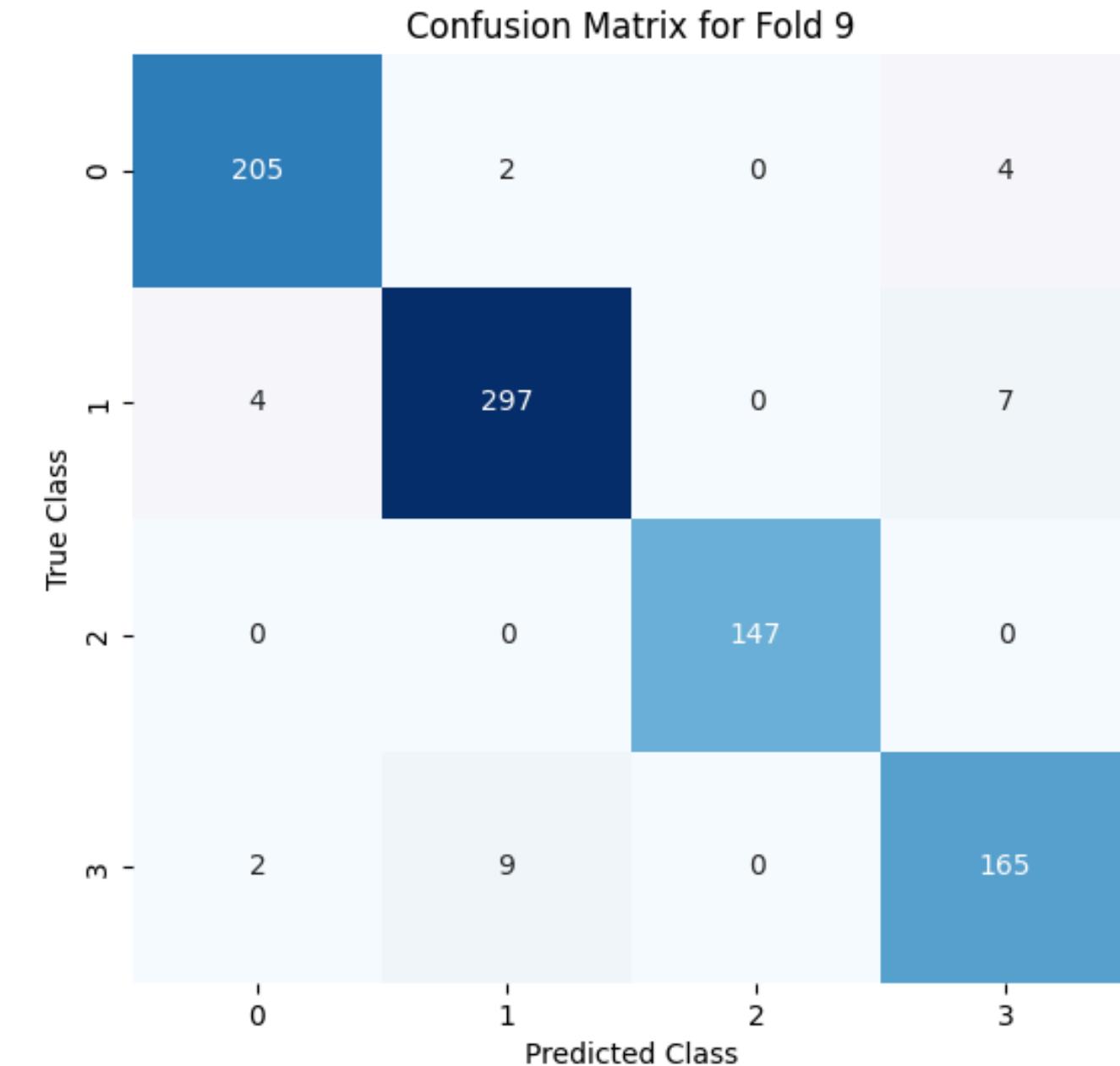
# 10-FOLD CROSS-VALIDATION (CONT.)



```

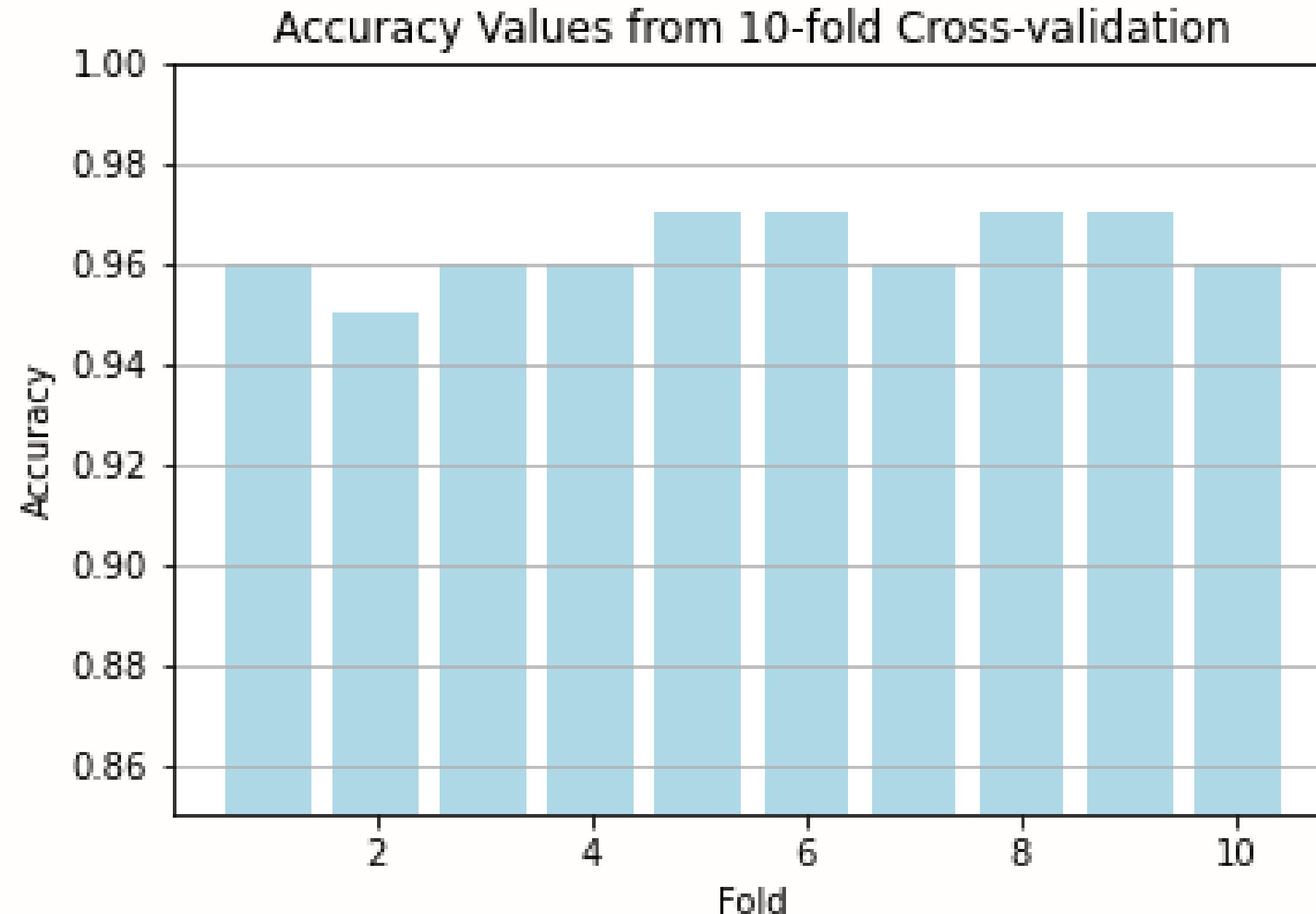
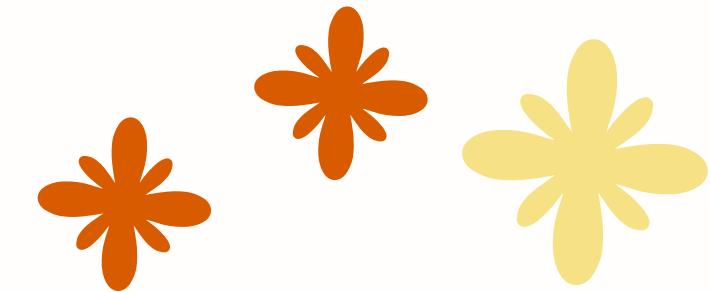
Fold 9:
Number of Images: 842
Correct Predictions: 814
Incorrect Predictions: 28
Accuracy: 0.967
Confusion Matrix:
[[205  2  0  4]
 [ 4 297  0  7]
 [ 0  0 147  0]
 [ 2   9  0 165]]
Classification Report:
      precision    recall  f1-score   support
0       0.97     0.97    0.97     211
1       0.96     0.96    0.96     308
2       1.00     1.00    1.00     147
3       0.94     0.94    0.94     176
accuracy                           0.97     842
macro avg    0.97     0.97    0.97     842
weighted avg  0.97     0.97    0.97     842

```

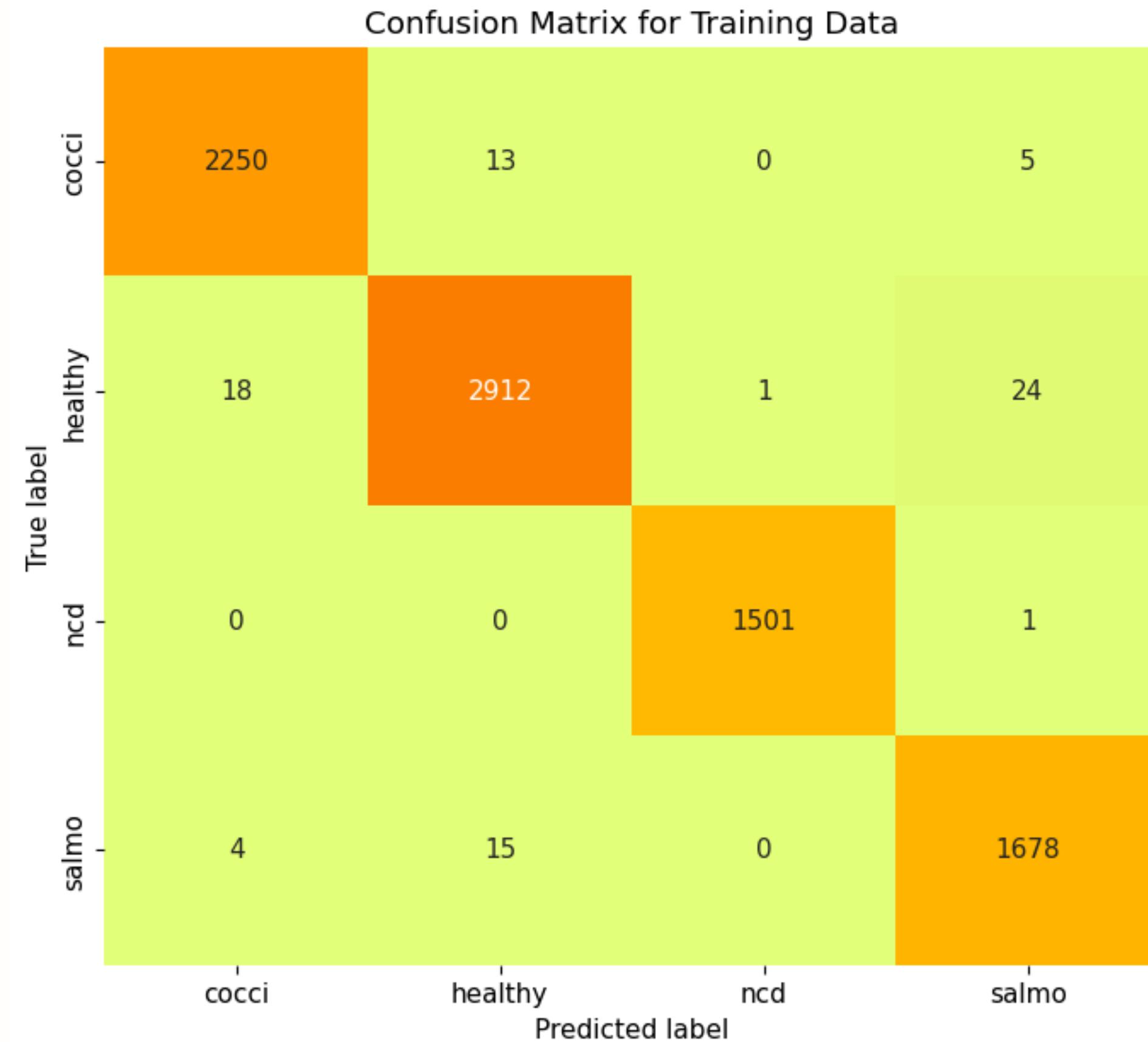
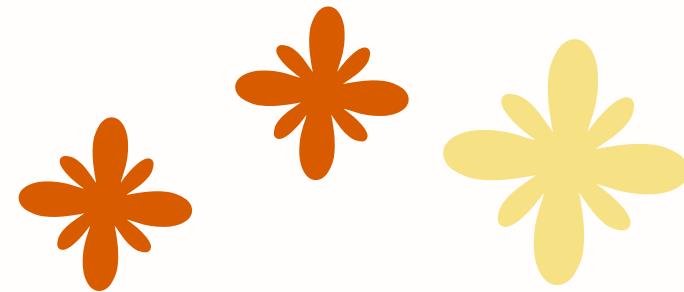


## EXAMPLE OF 10-FOLD CROSS-VALIDATION CALCULATION

# 10-FOLD CROSS-VALIDATION (CONT.)

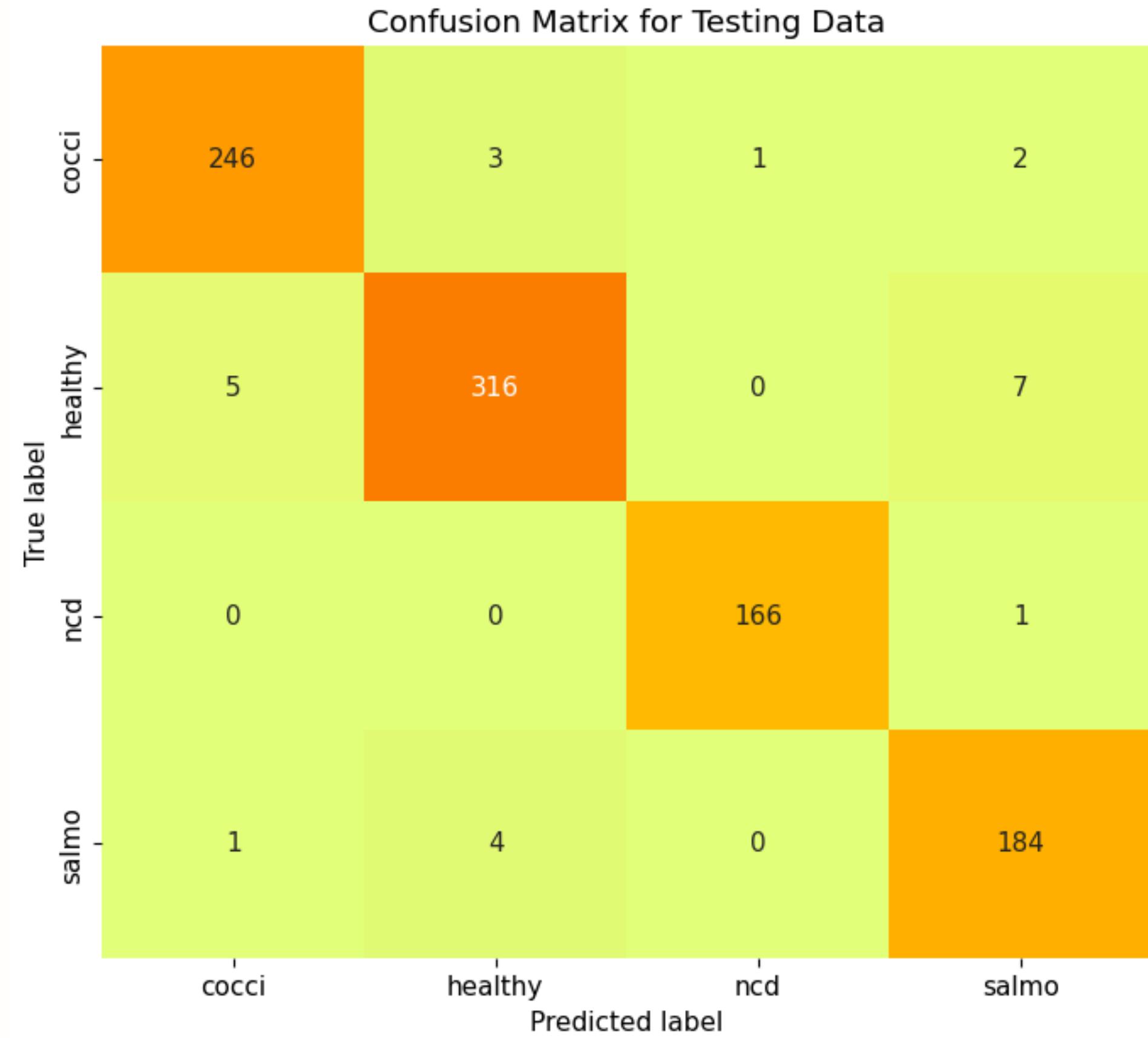
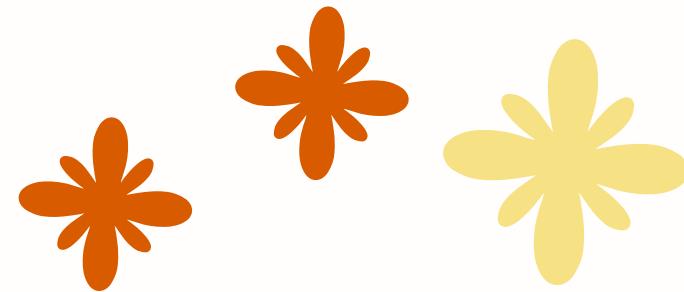


# FINAL MODEL TRAINING



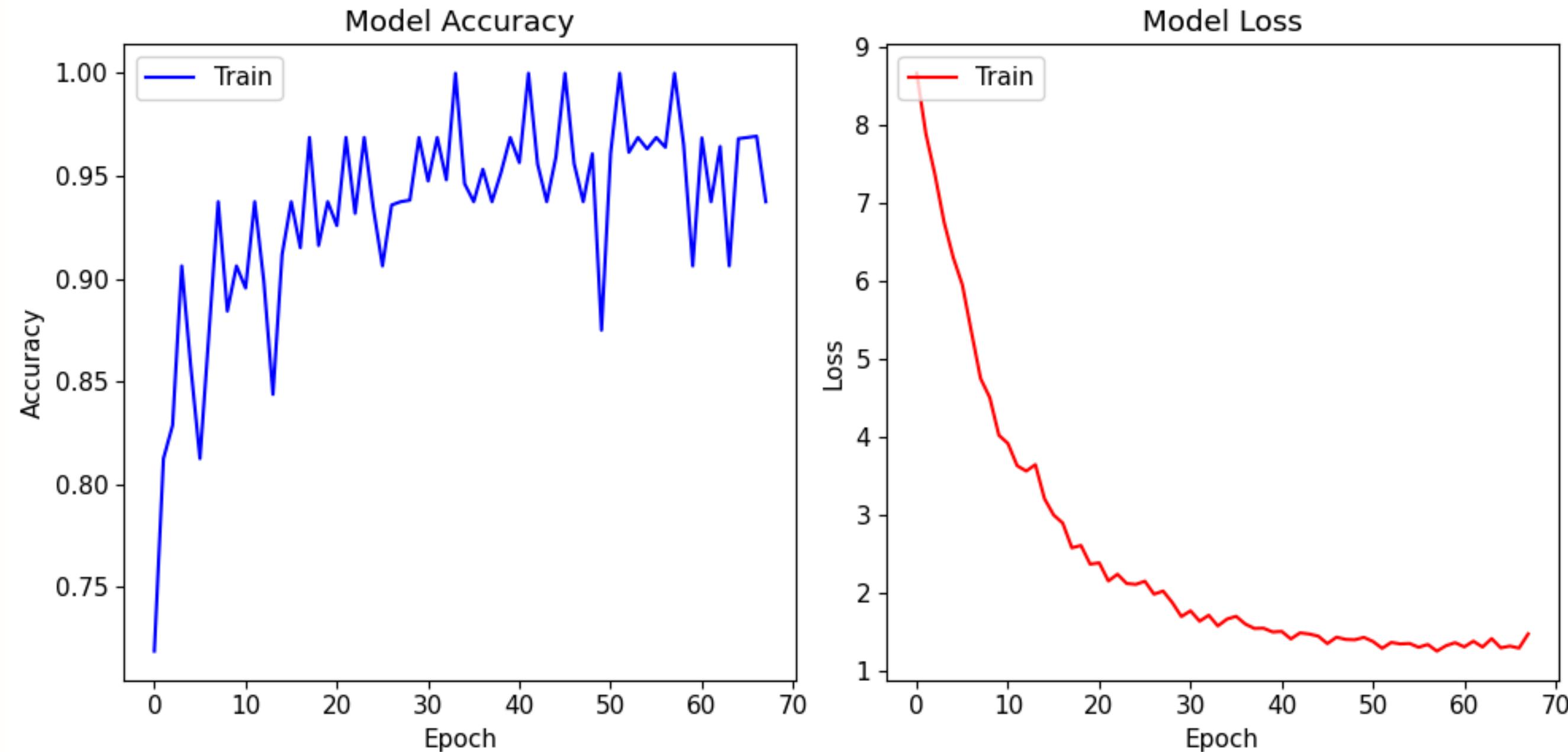
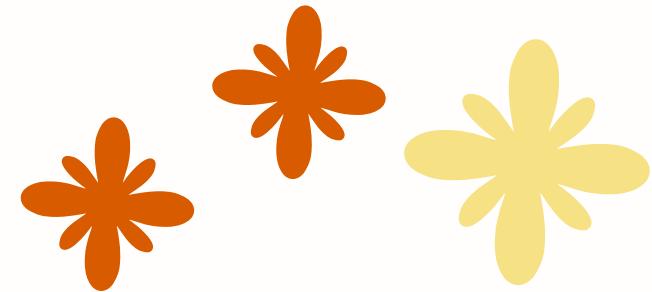
THE ACCURACY OF THE MODEL  
ON THE TRAINING DATA: 0.99

# FINAL MODEL TESTING



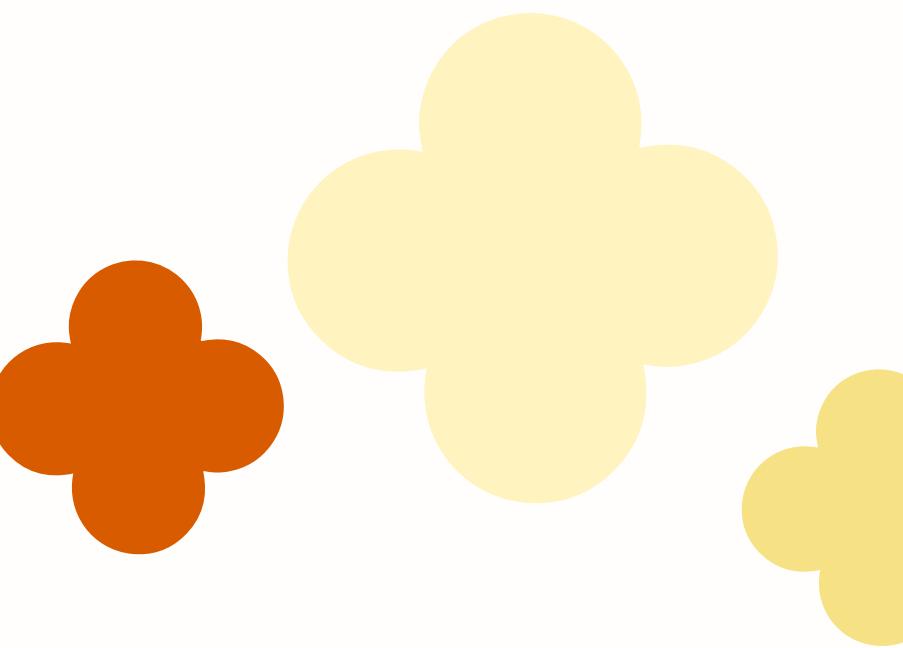
THE ACCURACY OF THE MODEL  
ON THE TESTING DATA: 0.97

# RESNET50 EXPERIMENT RESULT (CONT.)

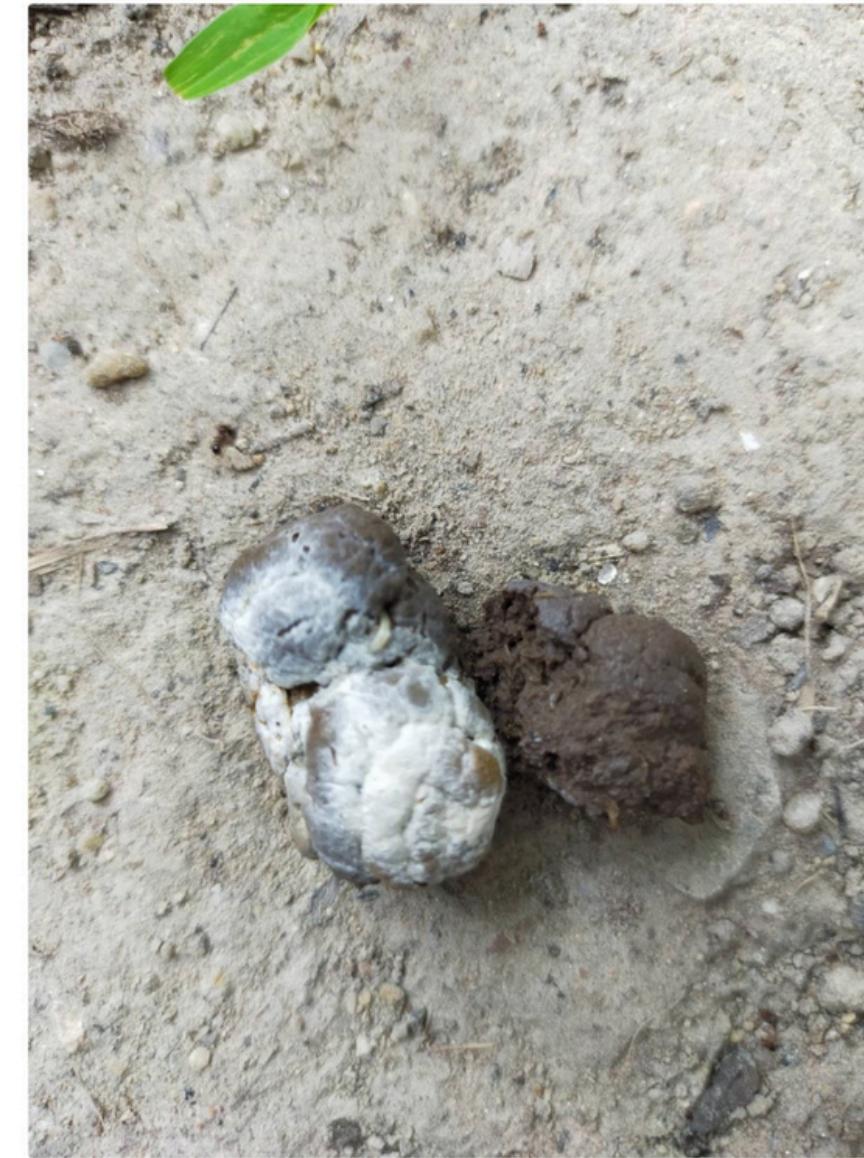
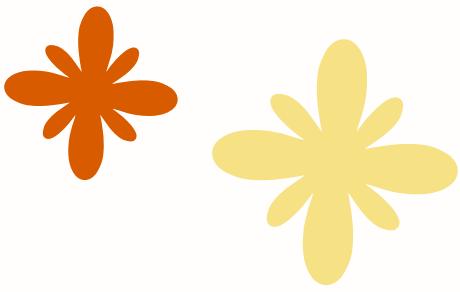


The designed ResNet50 model was trained on a dataset of labeled images using a 90/10 train-test split ratio over 100 epochs, achieving an accuracy of 99%.

# THAI DATASET

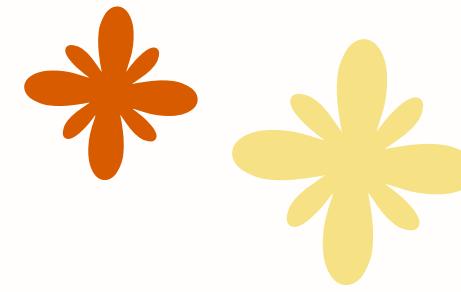


# THAI CHICKEN FECES IMAGES DATASET



- Chang Khwang, Kanchanadit District, Surat Thani Province, Thailand
- 561 chicken feces images, captured via a mobile phone camera by local poultry farmers in March 2024.

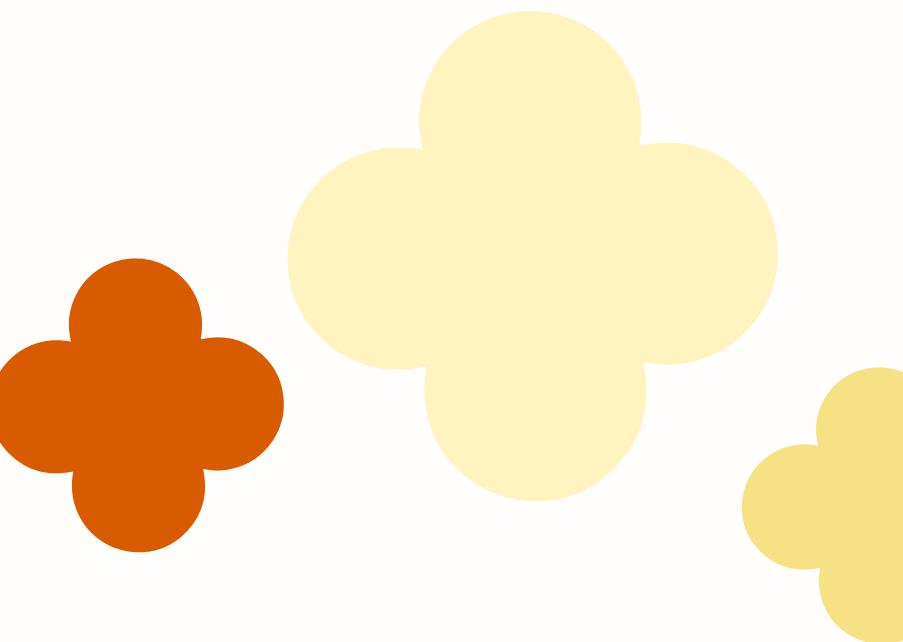
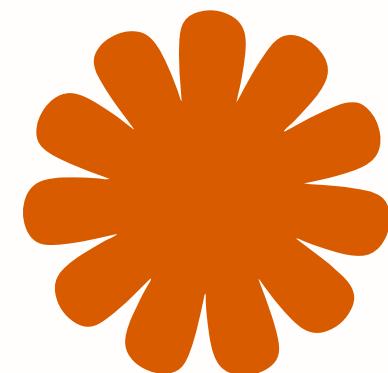
# EXPERIMENT WITH THAI CHICKEN FECES IMAGES



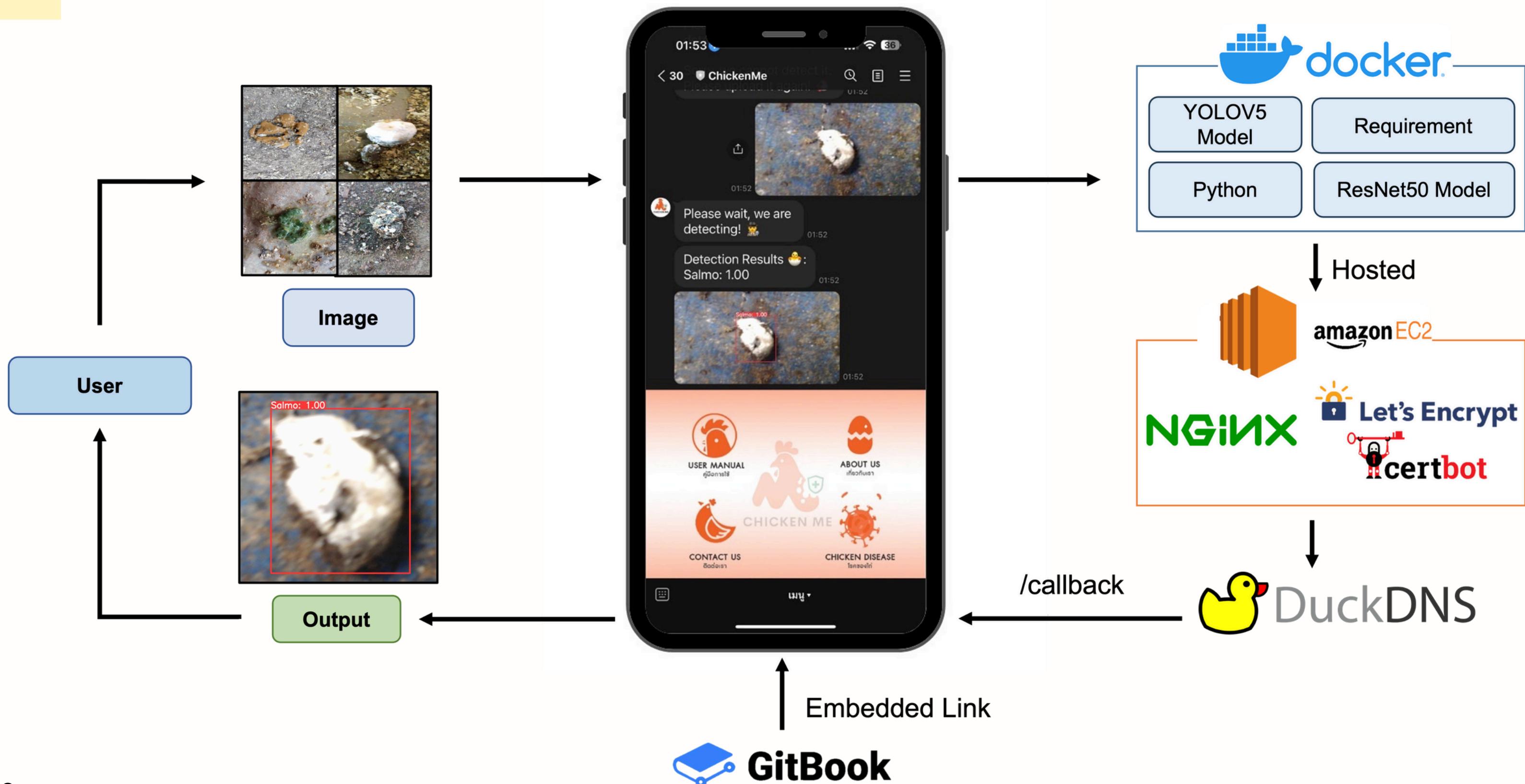
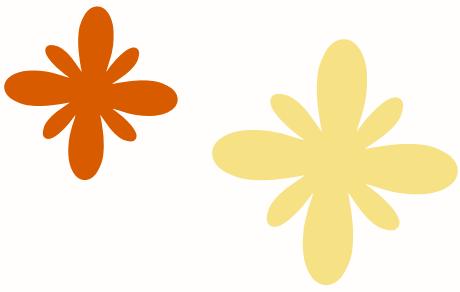
- Confirm with the farm owner that the Thai chicken is in good health.



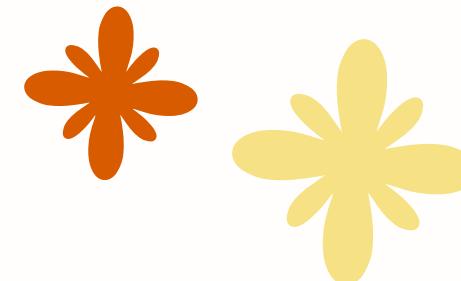
# **IMPLEMENTATION WITH LINE OA**



# IMPLEMENTATION WITH LINE OA



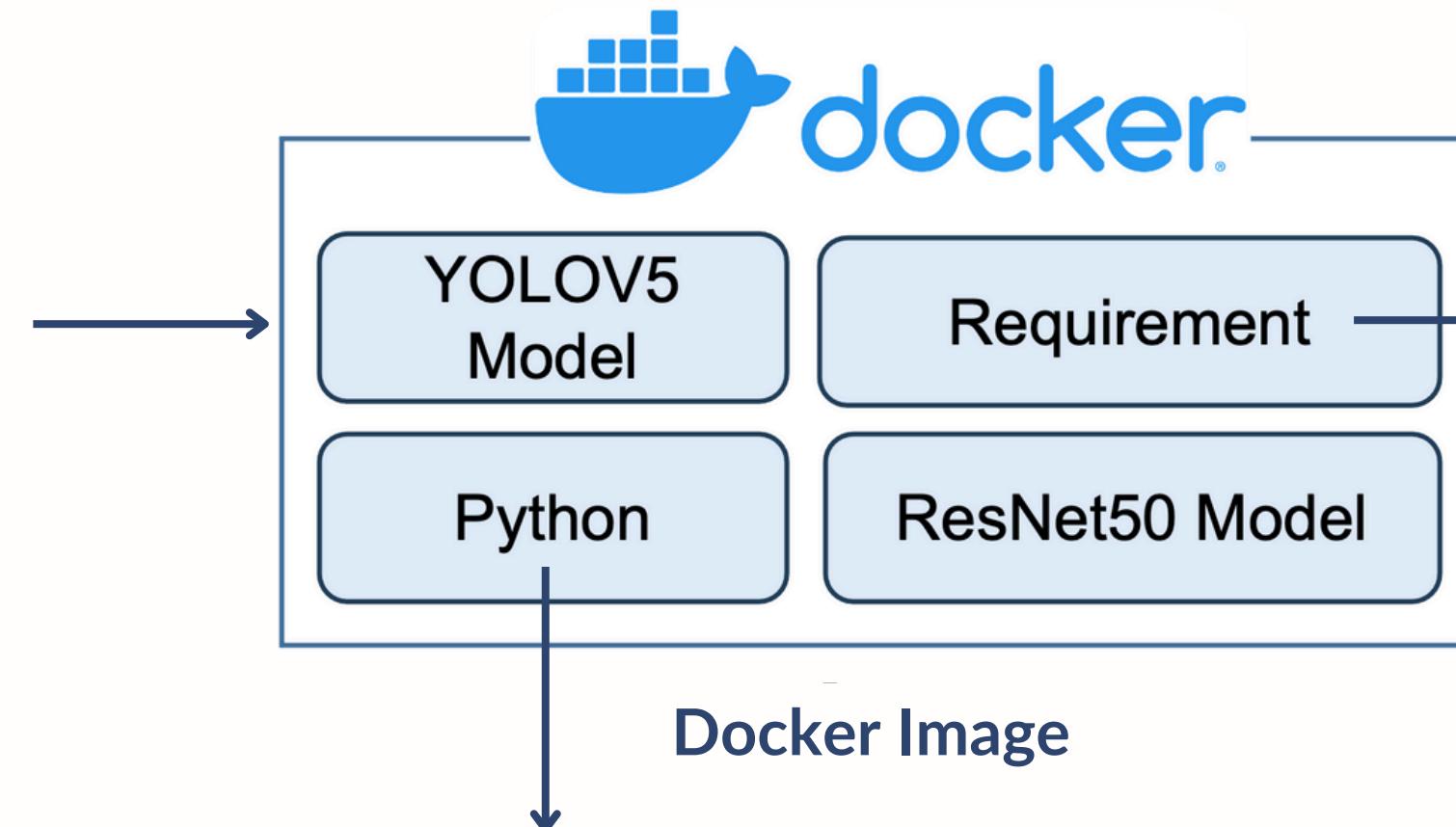
# IMPLEMENTATION WITH LINE (CONT.) \*



```

FROM python:3.10.13-slim
WORKDIR /app
RUN apt-get update && apt-get install -y \
    ffmpeg \
    libsm6 \
    libxext6 \
    && apt-get clean && rm -rf /var/lib/apt/lists/*
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY ..
ENV YOLO_CONFIG_DIR=/tmp/yolo
ENV MPLCONFIGDIR=/tmp/matplotlib
RUN echo "Make sure Torch and Flask are installed:" && \
    python -c "import flask" && \
    python -c "import torch"
CMD ["gunicorn", "line_up_fix:app", "--bind", "0.0.0.0:8000", "--workers", "1", "--timeout", "240"]
EXPOSE 8000
  
```

Dockerfile



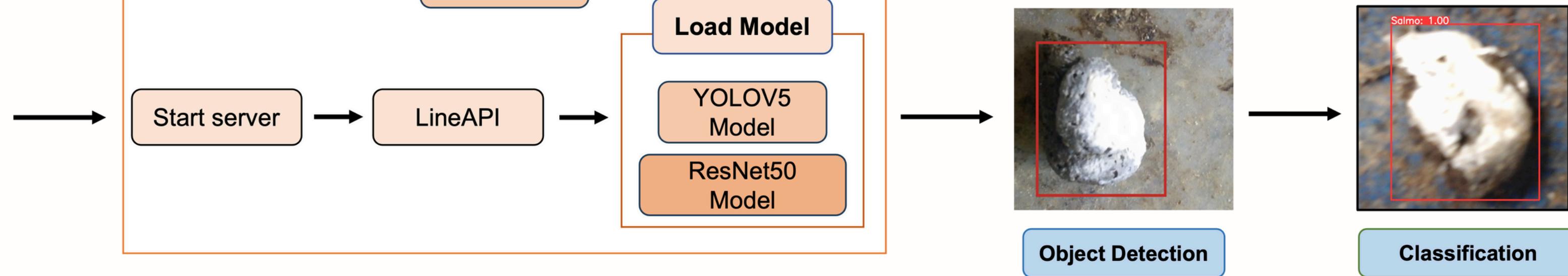
Requirements:

- Flask==3.0.2
- gunicorn==21.2.0
- line-bot-sdk==3.9.0
- tensorflow==2.16.1
- torch==2.2.0
- yolov5==7.0.13
- numpy==1.26.4

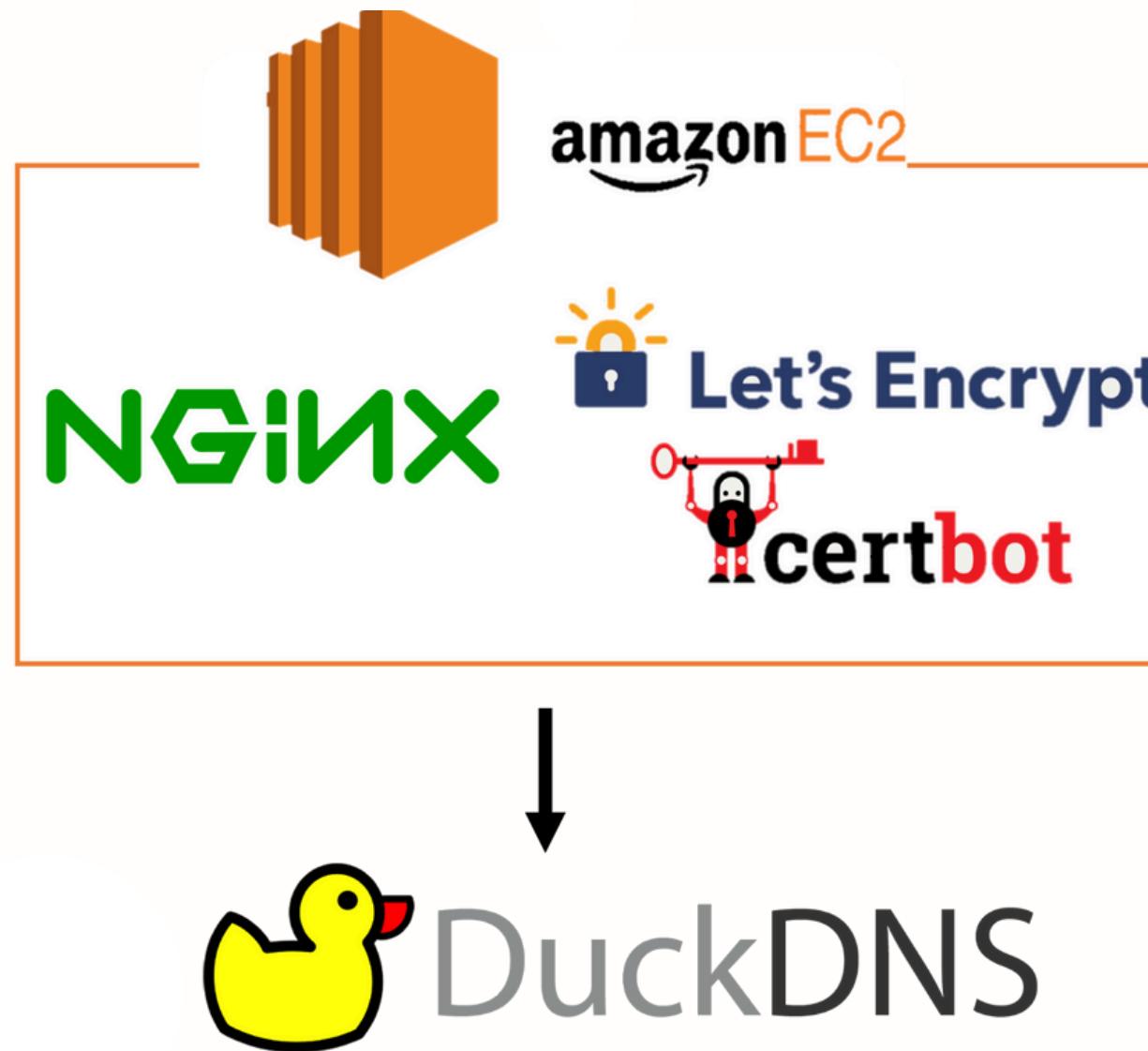
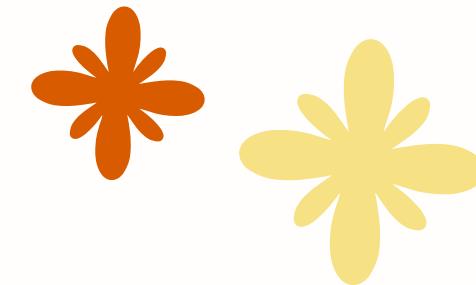
Docker Image



Image



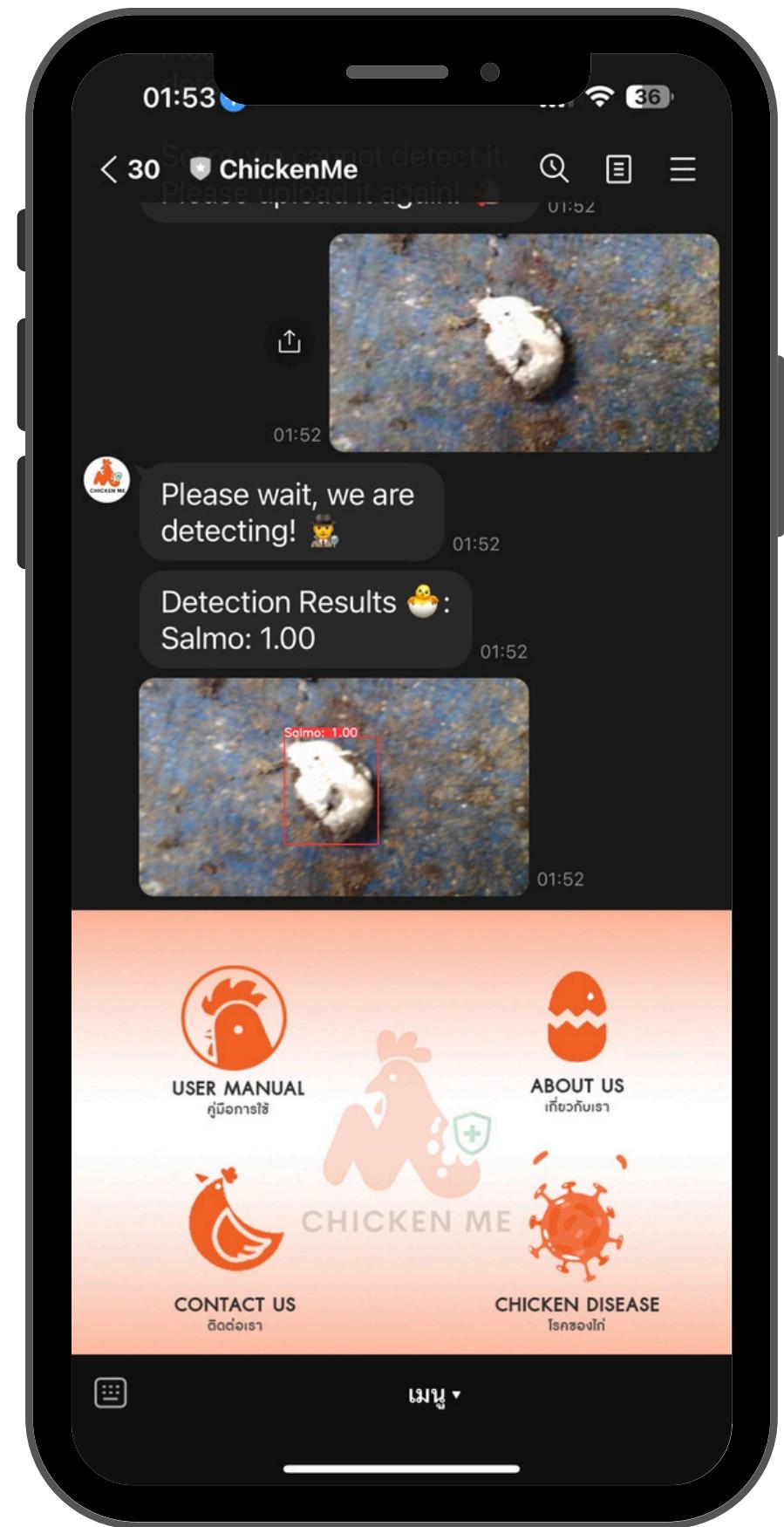
# IMPLEMENTATION WITH LINE (CONT.) \*



## Nginx Configuration for “chickenme.duckdns.org”

- HTTPS Configuration:
  - Port: 443 (SSL enabled)
  - SSL Certificates: Managed by Certbot (Auto-renewable from Let's Encrypt)
  - Proxy to Application: Requests are forwarded to an app running on localhost port 8000.
  - Header Settings: Pass client IP and protocol details to the backend for accurate request handling.
- HTTP to HTTPS Redirection:
  - Port: 80 (Standard for HTTP)
  - Behavior: All HTTP traffic is permanently redirected to HTTPS, ensuring secure connections.

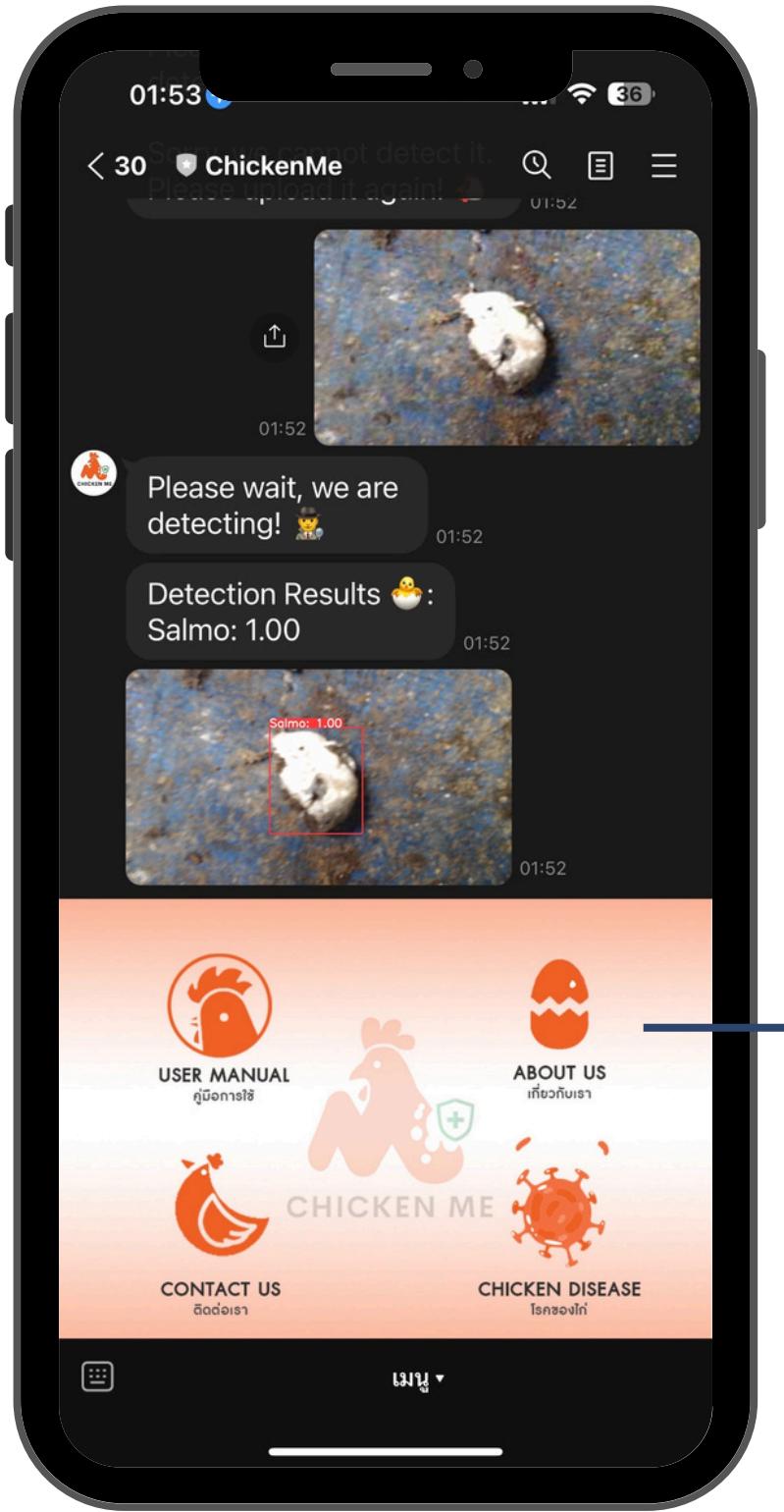
# ChickenME LINE ACCOUNT



## RICH MENU

- **User Manual:** User guidance on how to use the ChickenME LINE account
- **About Us:** Information about the project and team.
- **Contact Us:** For users who want to inquire or ask for support
- **Chicken Disease Information:** To help users gain a better understanding of diseases in chicken.

# ChickenME GITBOOK



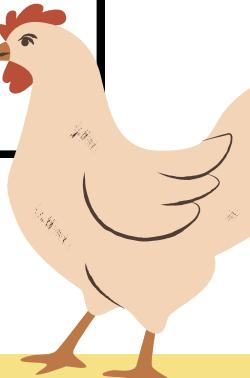
Embedded Link



**เกี่ยวกับเรา**

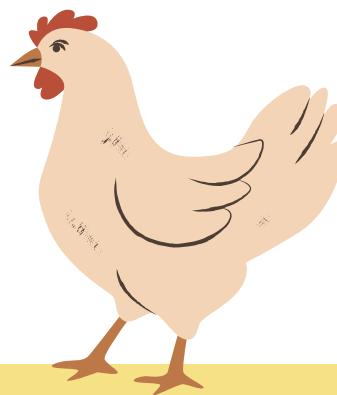
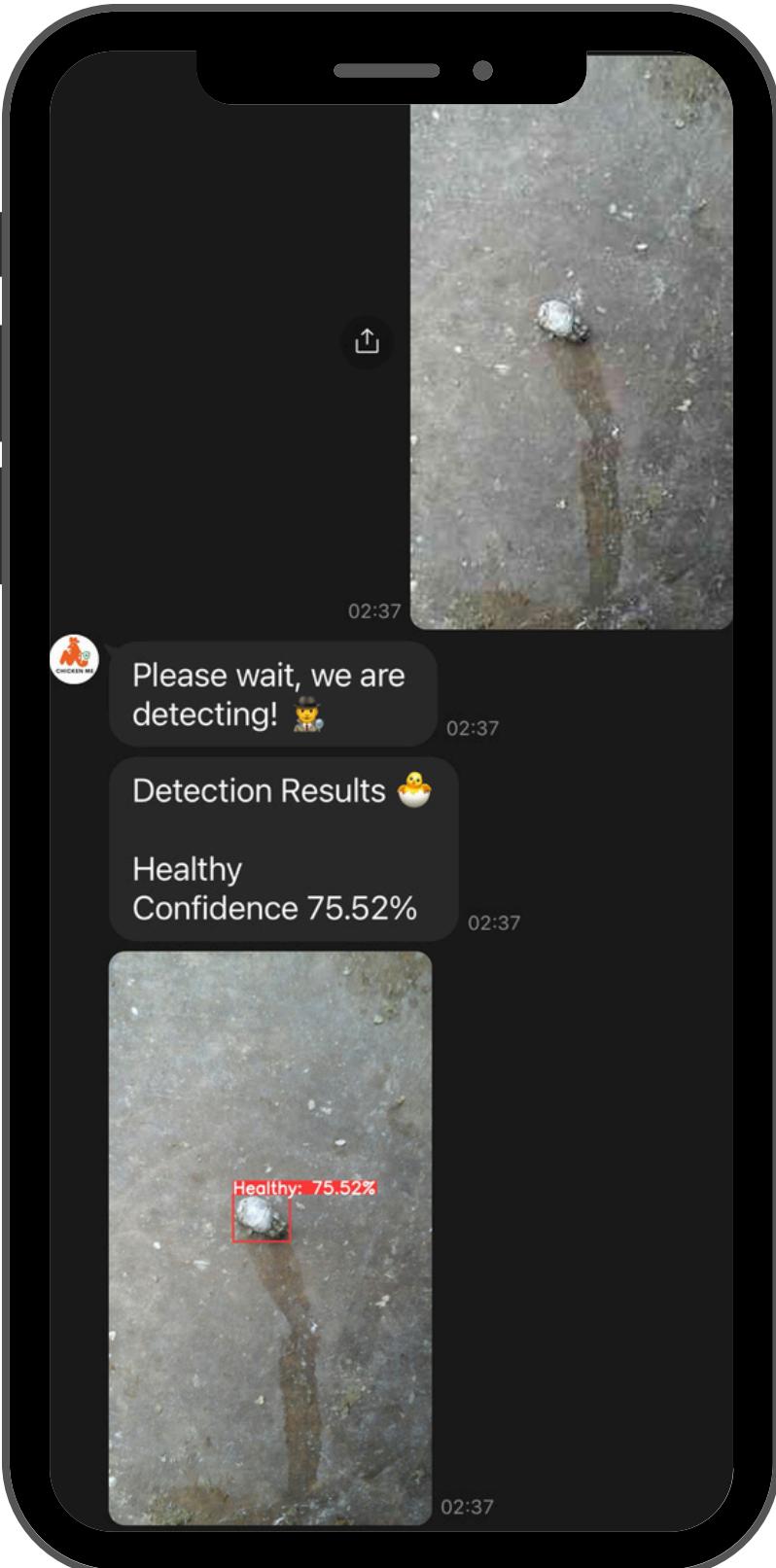
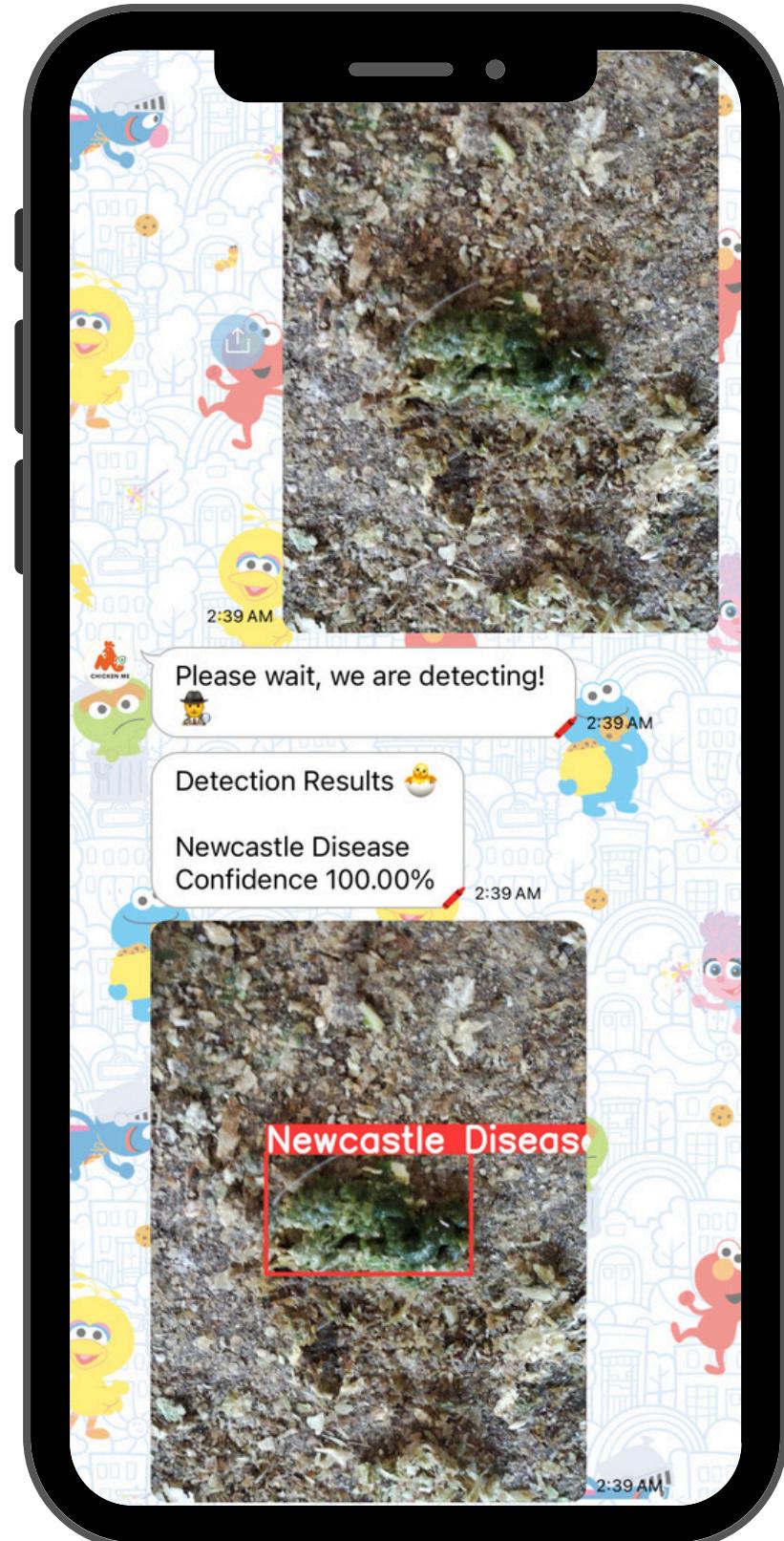
**ประวัติความเป็นมา**

โครงการ ChickenME เป็นเครื่องมือที่ใช้เทคโนโลยีการเรียนรู้เชิงลึกเพื่อเพิ่มประสิทธิภาพในการตรวจสอบและจัดการโรคในไก่ การเพิ่มเติมฟีเจอร์เพื่อเพิ่มประสิทธิภาพในการตรวจสอบโรคและลดการพึ่งพาต่อการสังเกตการณ์โดยมีเป้าหมายในการเพิ่มอัตราการตรวจพบอย่างมีประสิทธิภาพ โครงการนี้มีส่วนร่วมในการสร้างความยั่งยืนในการเลี้ยงดูไก่และสนับสนุนเป้าหมาย Zero Hunger ผ่านการปฏิบัติทางการเกษตรที่ยั่งยืน นอกจากนี้ โครงการยังใช้แพลตฟอร์ม Line Official Account เพื่อส่งเสริมการใช้งานในกลุ่มเกษตรกรและคนเลี้ยงไก่ นอกเหนือจากการสร้างความยั่งยืนในการอาหารและความสัมพันธ์ระหว่างมนุษย์กับไก่แล้ว โครงการยังมุ่งเน้นการเพิ่มศักยภาพในการพัฒนาและเสริมสร้างสุขภาพของไก่ในภารกิจการเกษตร ดังนั้น โครงการ ChickenME เป็นเครื่องมือที่มีความสำคัญในการพัฒนาและเสริมสร้างสุขภาพของไก่และในการสนับสนุนการพัฒนาเศรษฐกิจภูมิภาคในทางเกษตรได้อย่างยั่งยืน และเป็นส่วนหนึ่งของการประสานงานสู่การบรรลุเป้าหมาย Zero Hunger ของชุมชนโลก



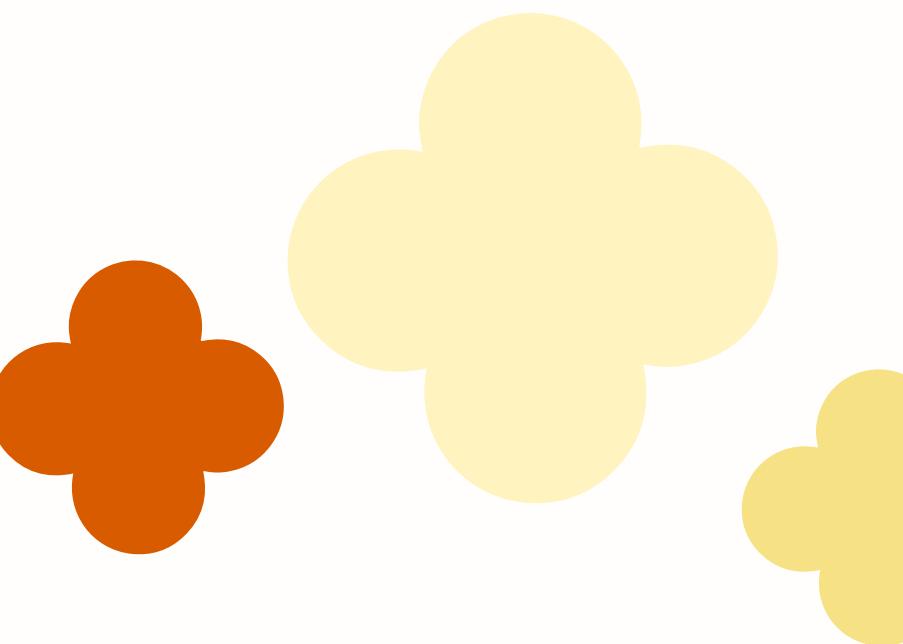
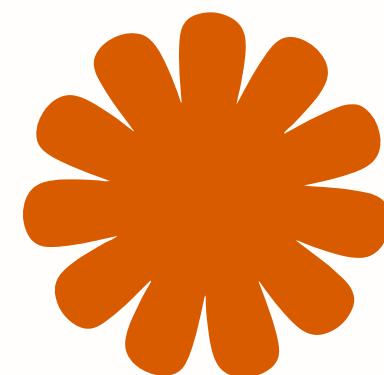
# ChickenME LINE ACCOUNT

Example images of processed output that users will receive from ChickenME





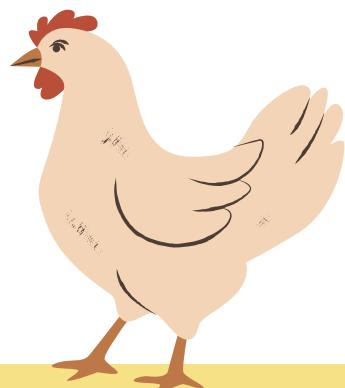
# CONCLUSION AND DISCUSSION





# CONCLUSION

- ChickenME utilized YOLOv5 for object detection and ResNet50 for image classification to classify the diseases in chickens.
- Classification of health conditions into healthy and three common diseases: Coccidiosis, Salmonella, and Newcastle Disease.
- Trained on a dataset of 10,500 chicken fecal images from Zenodo, ChickenME automates disease detection. With a split ratio of 90/10.
- From the experiments, YOLOv5 gave a better result than YOLOv3 when training on the same hyperparameter and settings.
- Integrated within the LINE platform, it offers user-friendly access and aims to decrease manual observation for time-efficient deployment.



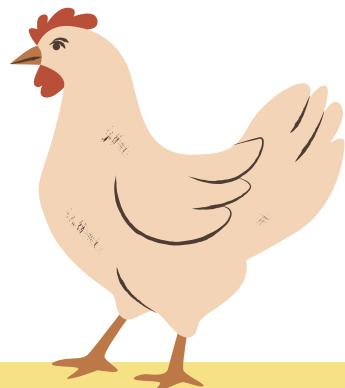


# LIMITATIONS

- ChickenME's current limitation is its limited disease classification range. It identifies common diseases but may miss other significant poultry diseases.
  - lead to undetected conditions affecting poultry health and farm productivity.
- Hardware and resource limitations affect model development
  - Restricted computational power
  - Limited memory and slower processing times

# FUTURE WORK

- The enhancement of ChickenME aims to expand disease classification beyond current inclusions.





# ChickenME

# DEMONSTRATION



# ChickenME

## DEMONSTRATION





# THANK YOU

Q&A session