



Early Detection of Digital Dermatitis in Dairy and Beef Cattle using Artificial Intelligence

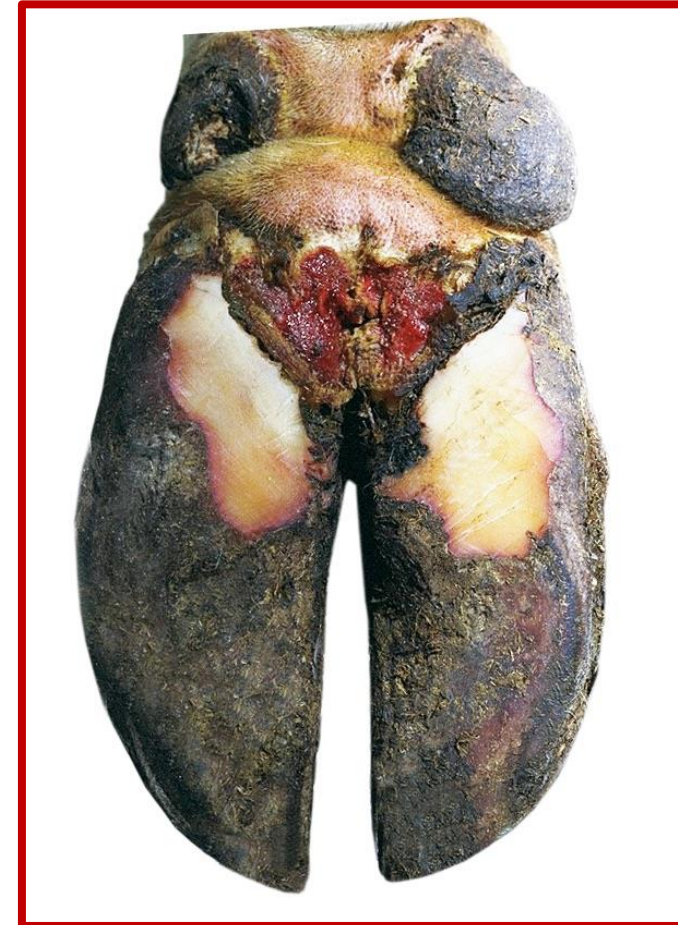
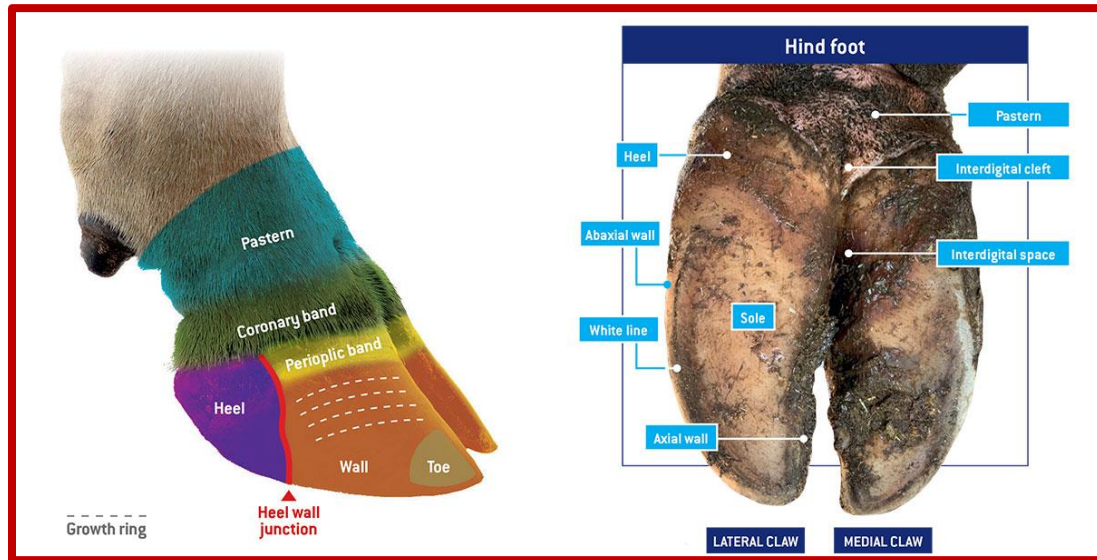
Srikanth Aravamuthan

Dr. Dörte Döpfer Laboratory

Department of Medical Sciences, School of Veterinary Medicine,
University of Wisconsin, Madison 53706

aravamuthan@wisc.edu

Digital Dermatitis

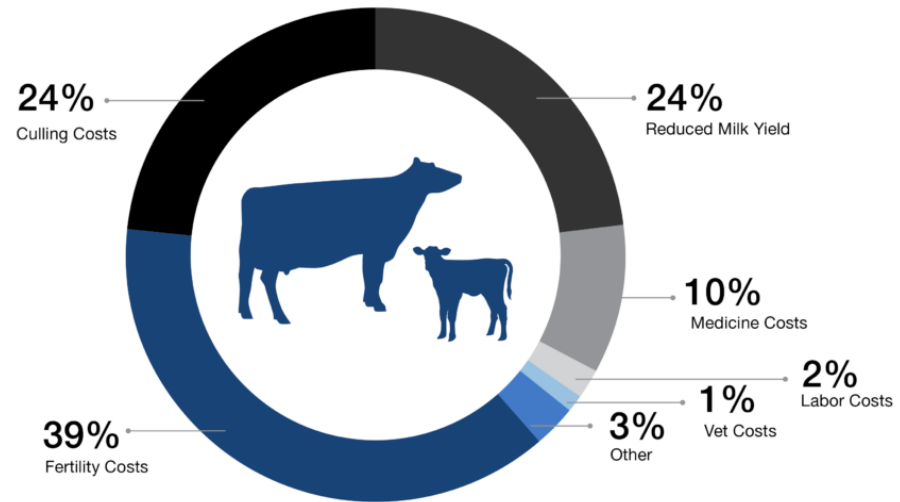


M-Stage Classification System



Lameness

ANNUAL COST OF LAMENESS FOR A DAIRY



Adapted from Willshire et al., 2009





Artificial Intelligence

- Artificial intelligence (AI) is used to perform a variety of advanced functions
 - Ability to see
 - Understand and translate spoken and written language
 - Analyze data
 - Make recommendations



Artificial Intelligence

 [Overview](#) [Documentation](#) [Examples](#) [Playground](#)

[Upgrade](#) [Help](#)  University of Wisconsin-Madison

Get started

Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

KEEP IN MIND

- Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more.](#)
- Requests submitted to our API will not be used to train or improve future models. [Learn more.](#)
- Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

Playground

Chat

Save View code Share

The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly.

Human: Hello, who are you?
AI: I am an AI created by OpenAI. How can I help you today?
Human: Can you give me presentation for me?
AI: Unfortunately, I'm not able to give presentations for you. However, I can help you by providing tips and resources that may assist you in creating the presentation yourself.
Human: No, thanks...

Looking for ChatGPT? [Try it now](#)

Submit ↶ ↷ ↺

106

Mode

Complete

Model

text-davinci-003

Temperature

0.9

Maximum length

150

Stop sequences

Enter sequence and press Tab

Human: X

AI: X

Top P

1

Frequency penalty

0

Presence penalty

0.6

Best of

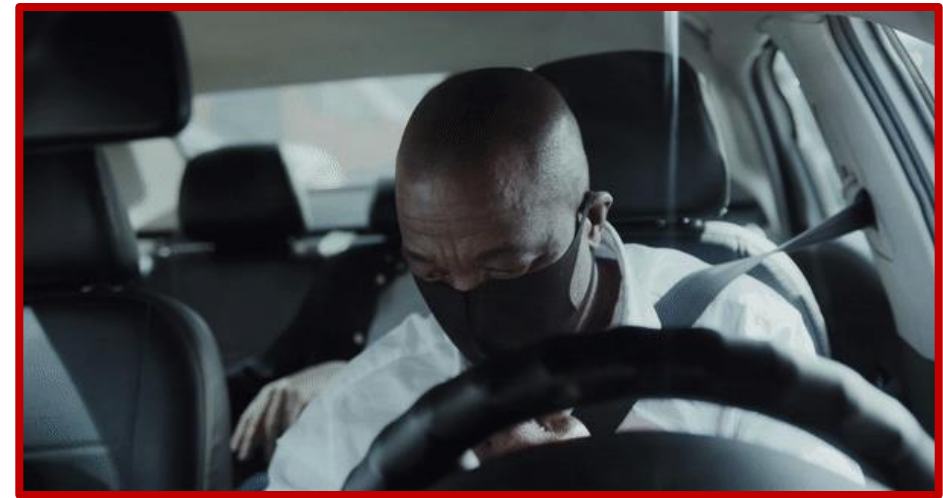
1

Inject start text

✓ #AT-

Computer Vision

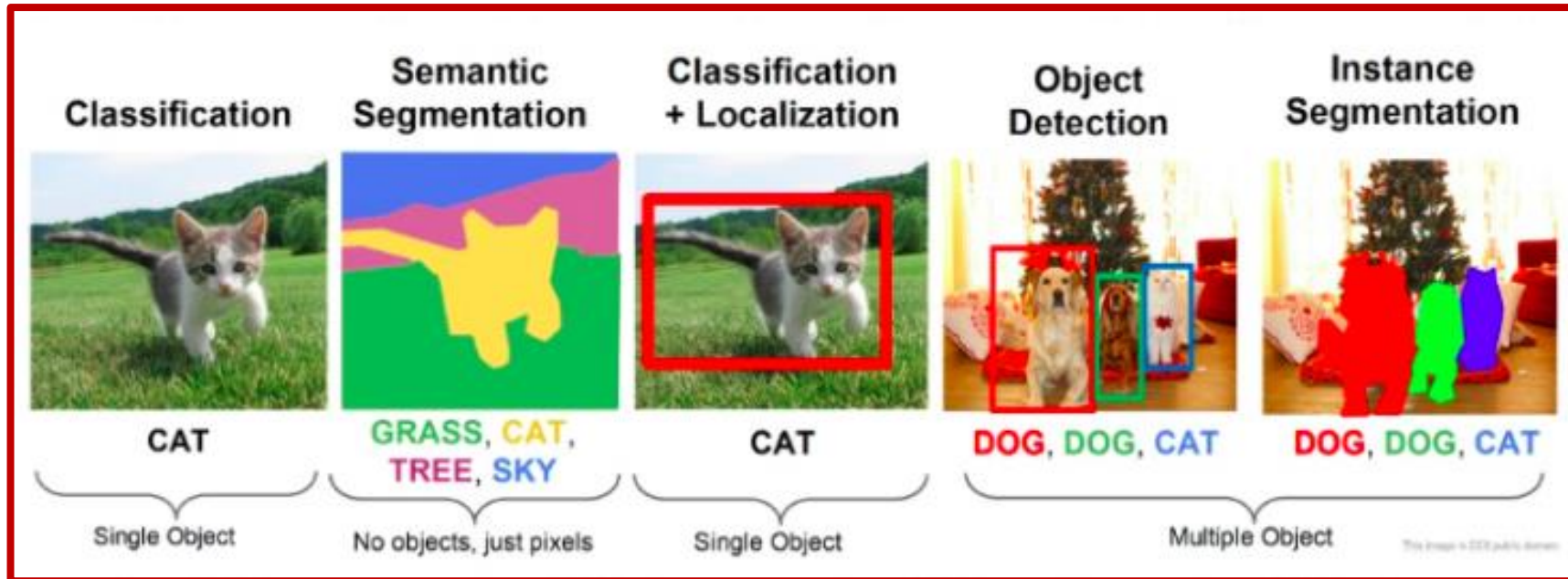
- Computer vision (CV) is used to acquire, process, analyze, and interpret images and videos.
 - Helps systems see and identify objects



Computer Vision

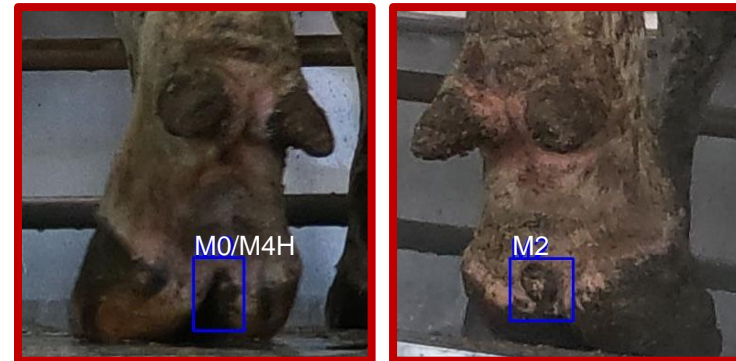


Computer Vision

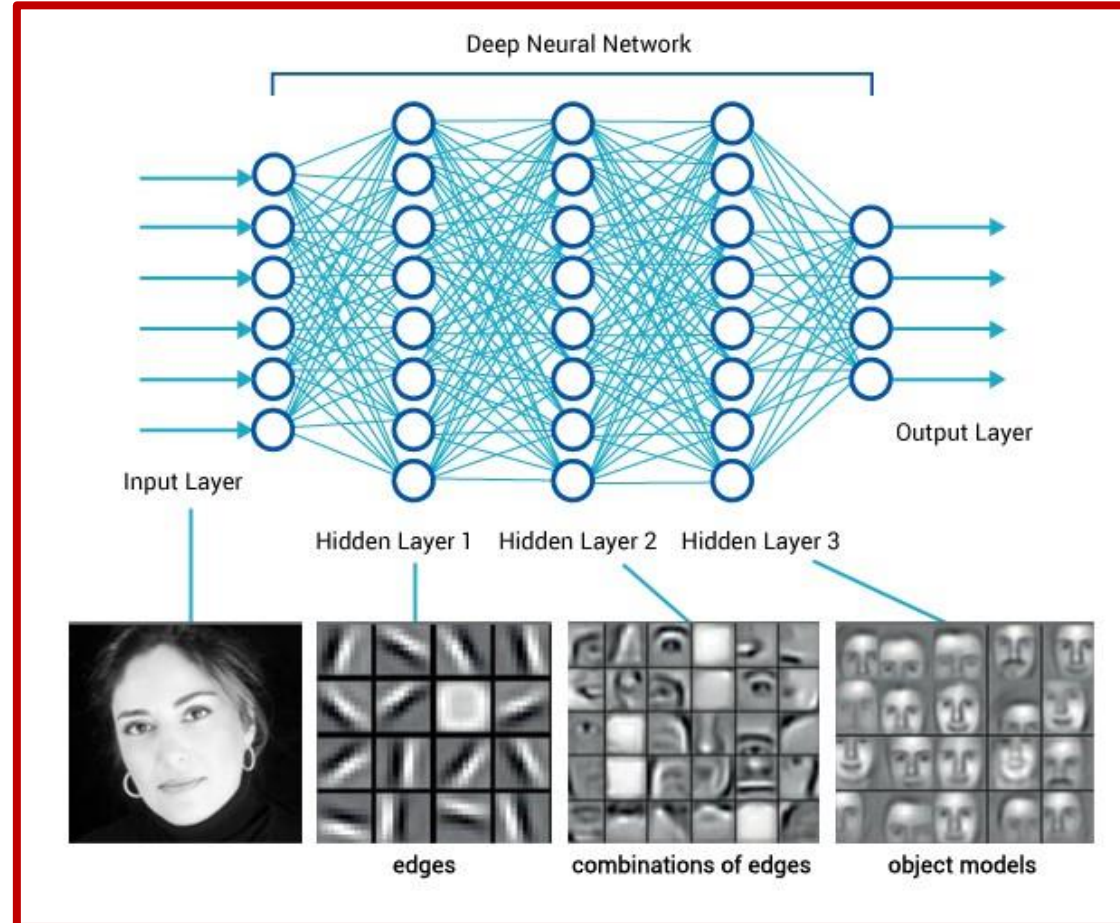


Object Detection

- Object detection locates the presence of objects with a **bounding box** and **class labels** of the located objects in an image.
 - Two-stage object detectors
 - Region-Based Convolutional Neural Networks (R-CNNs)
 - One-stage object detectors
 - Single-Shot Detectors (SSD)
 - You Only Look Once (YOLO)

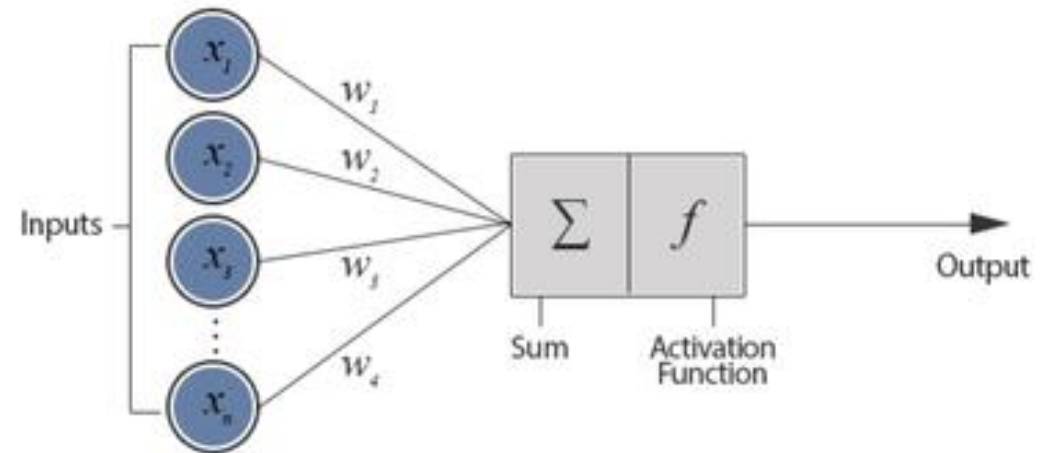
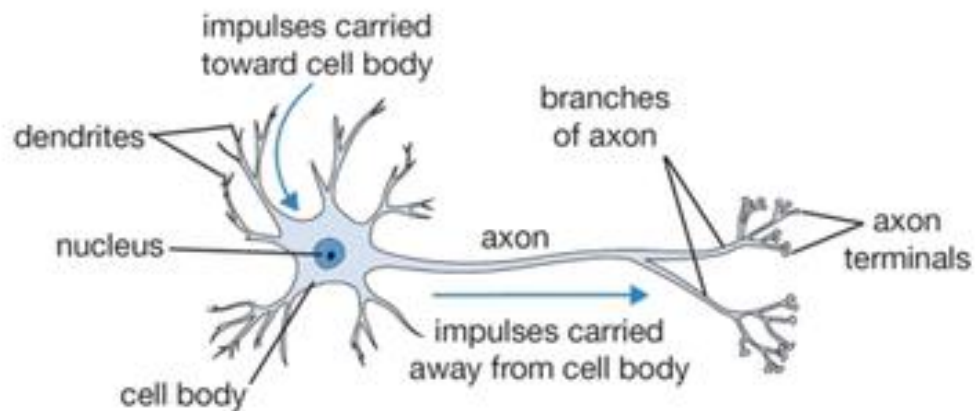


Neural Network (NN)



Neural Network (NN)

Biological Neuron versus Artificial Neural Network



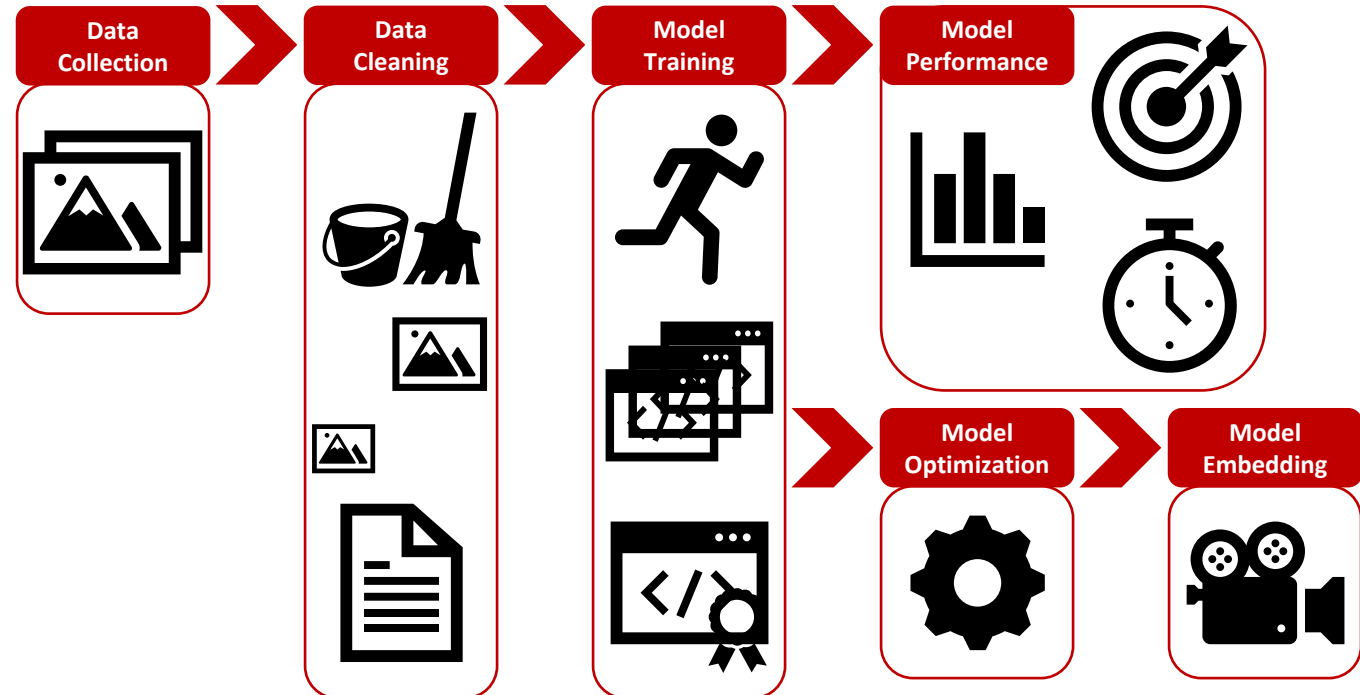
Motivation

- The purpose of this project is to implement a CV model for the **real-time detection of DD** in dairy and beef cattle.
- The motivation is to **minimize the effects of DD-associated lameness** in all cattle by means of early detection, prevention, and prompt treatments.



Approach

1. Data Collection
2. Data Cleaning
3. Model Training
4. Model Performance
5. Model Optimization
6. Model Embedding



Workflow



Data Collection

- Camera facing the backside of the hind foot with a clear view of the interdigital space of the hoof
- Two sets of images
 - 2,227 JPEG images of **single lesion** for Dataset 1
 - 409 JPEG images of **multiple lesions** for Dataset 2
- Scored for M-stages of DD by a trained investigator
 - **M0/M4 and M2** for Dataset 1
 - **M0, M2, M2P, M4H, and M4P** for Dataset 2

Workflow

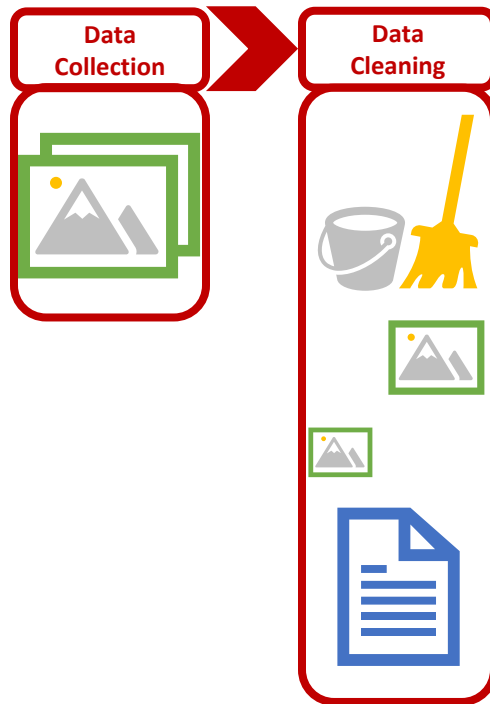
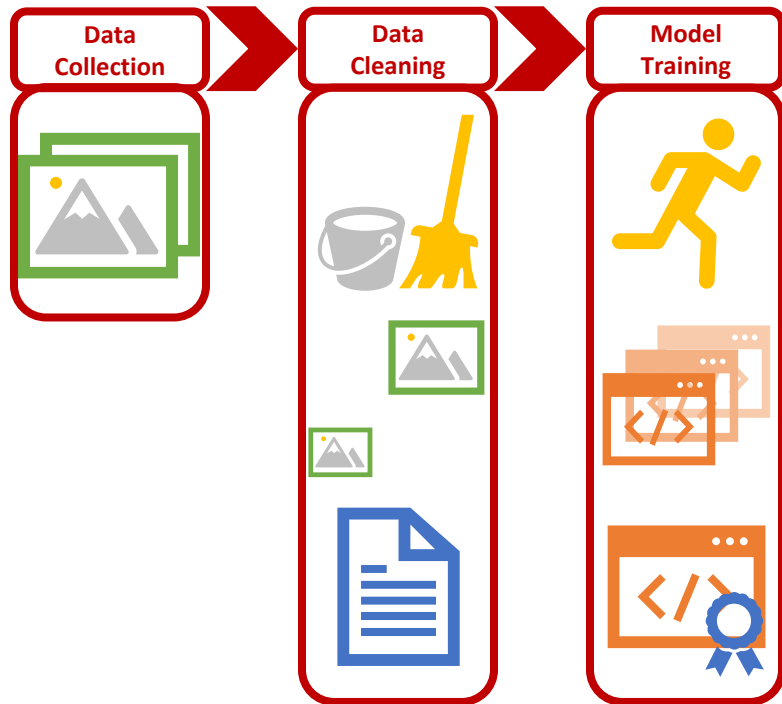


Image Labeling and Data Processing



Workflow



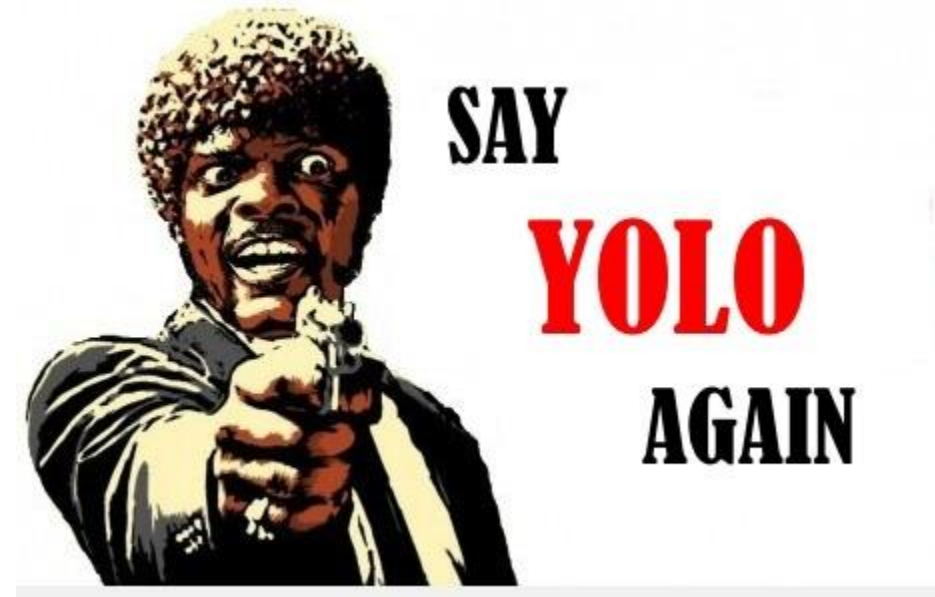
Model Training

- 90% images for training and 10% images for testing
- 8 object detection models
 - Faster R-CNN & Cascade R-CNN
 - SSD & SSD Lite
 - YOLOv3 & Tiny YOLOv3
 - YOLOv4 & Tiny YOLOv4

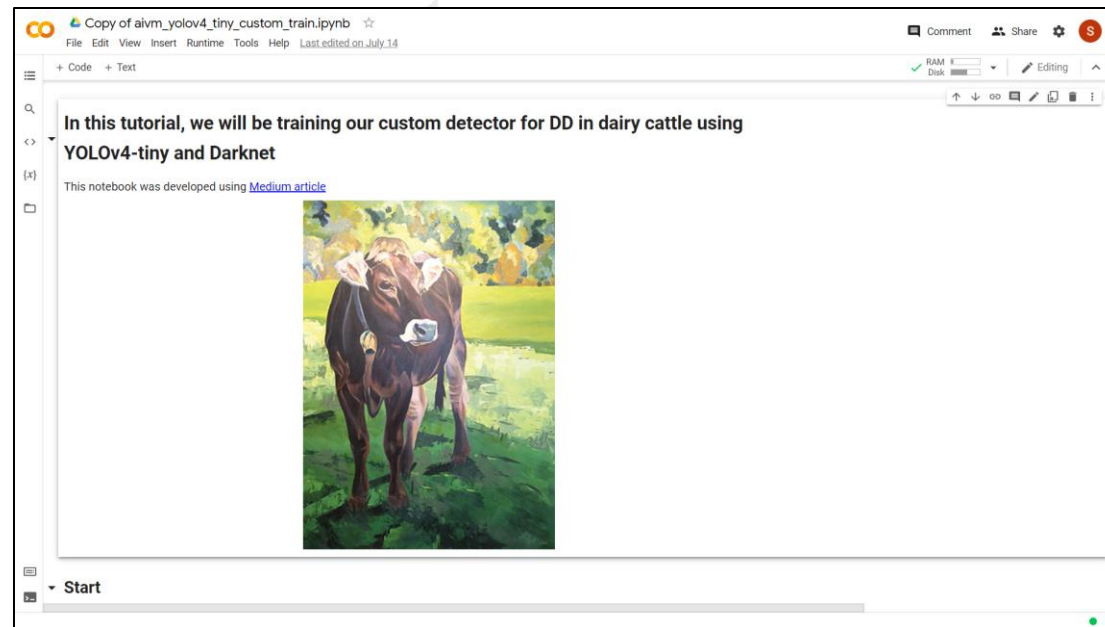


Model Training

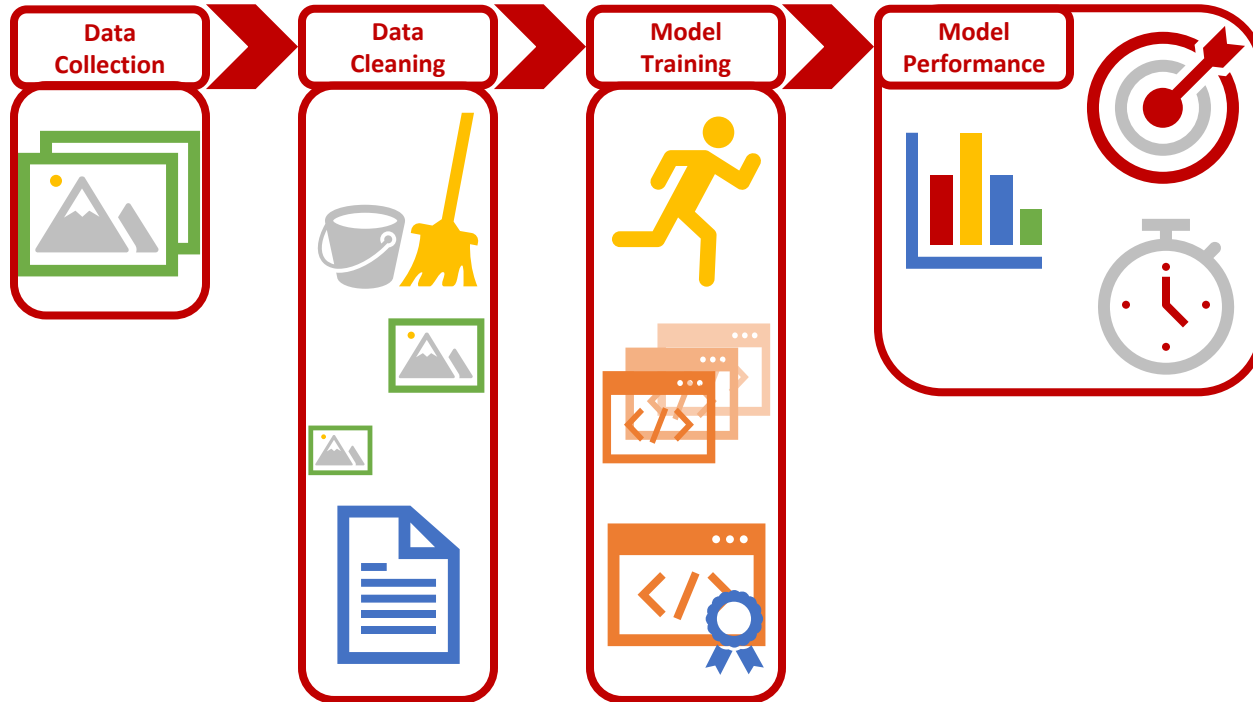
- 90% images for training and 10% images for testing
- 8 object detection models
 - Faster R-CNN & Cascade R-CNN
 - SSD & SSD Lite
 - YOLOv3 & Tiny YOLOv3
 - YOLOv4 & Tiny YOLOv4



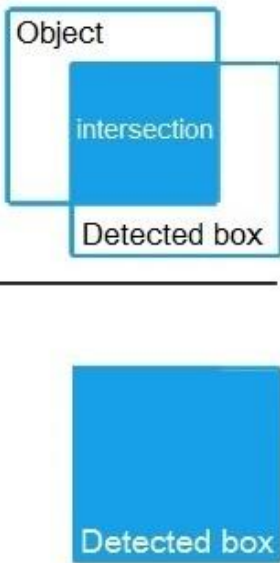
Model Training

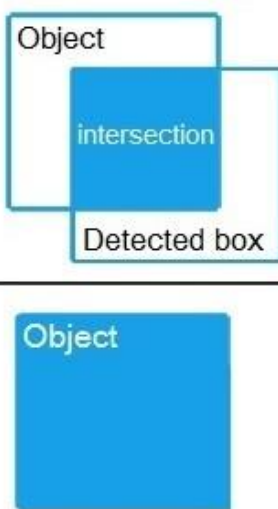


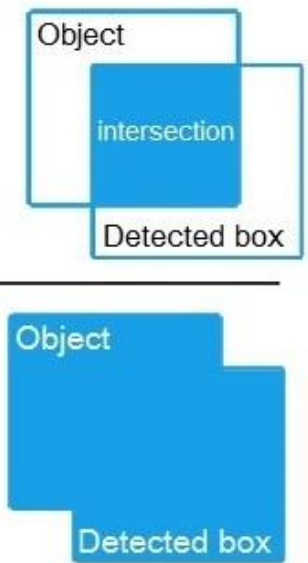
Workflow



Model Evaluation

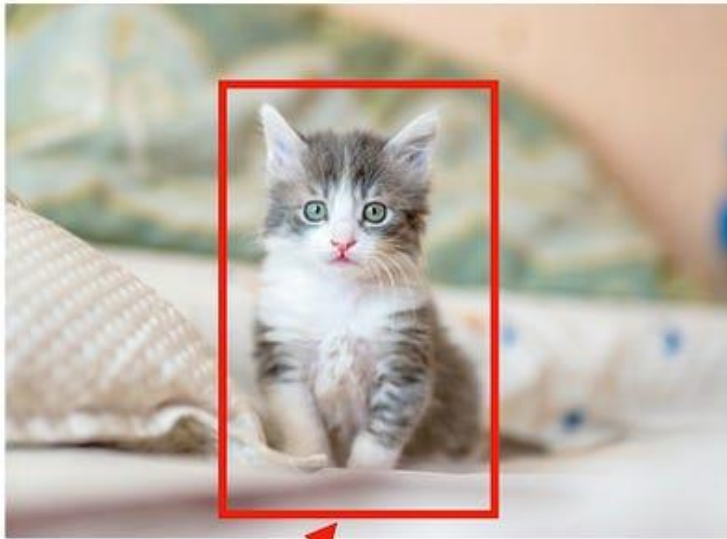
$$\text{Precision} = \frac{\text{Area of Intersection}}{\text{Area of Detected box}}$$


$$\text{Recall} = \frac{\text{Area of Intersection}}{\text{Area of Object}}$$


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Model Evaluation

Localisation



Here is the CAT

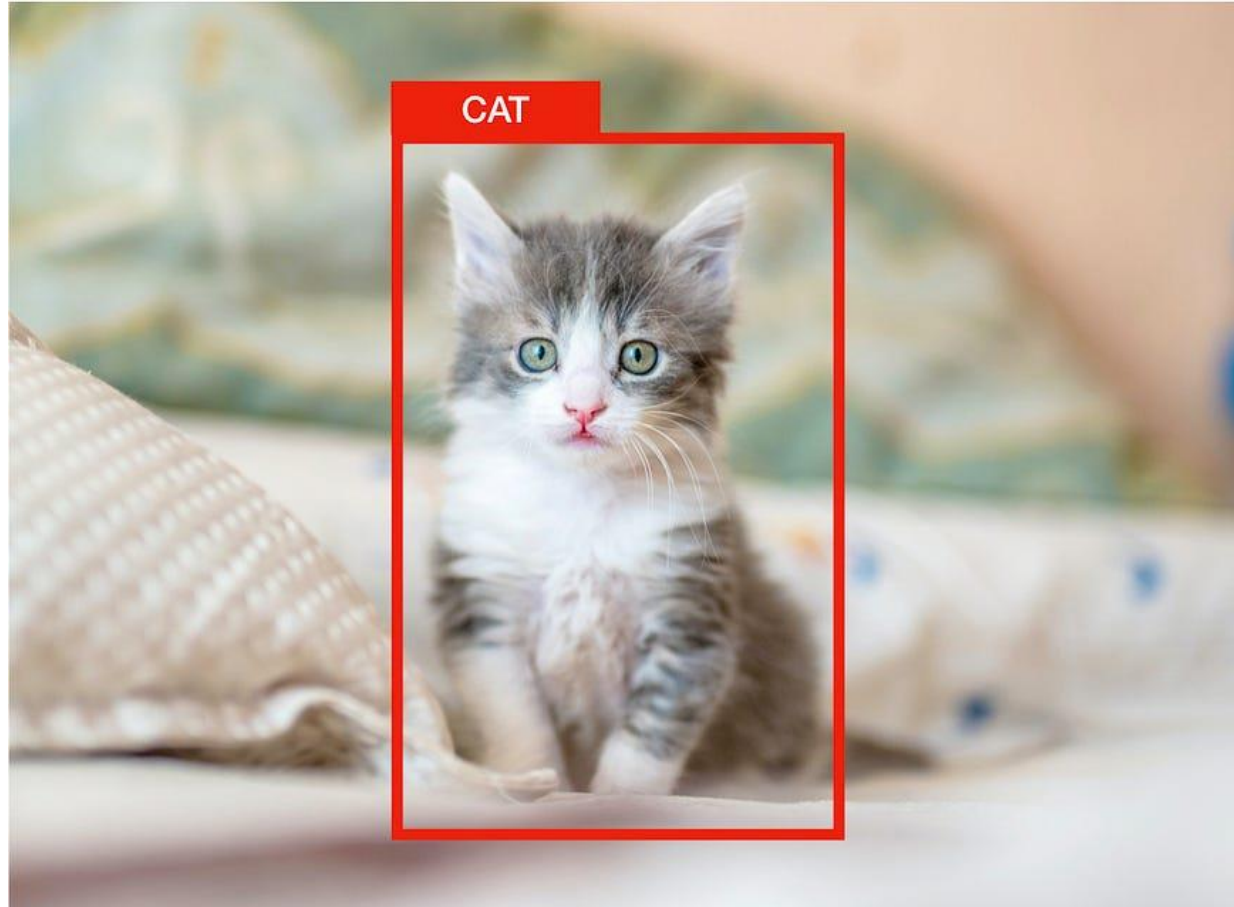
Cat image. Photo by Rola Puerto on Unsplash

Classification

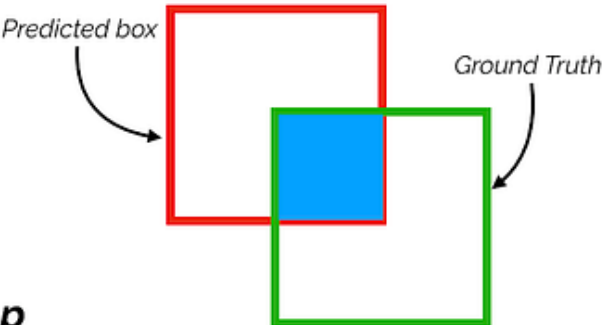


This is an image of CAT

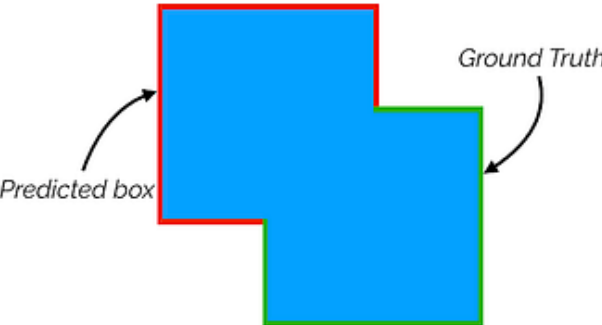
Model Evaluation



Model Evaluation

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Area of Overlap}}{\text{Area of Predicted box} + \text{Area of Ground Truth} - \text{Area of Overlap}}$$


The diagram shows two overlapping rectangles. The top rectangle is outlined in red and labeled 'Predicted box'. The bottom rectangle is outlined in green and labeled 'Ground Truth'. The intersection of the two rectangles is filled with blue. Arrows point from the labels to their respective boxes.

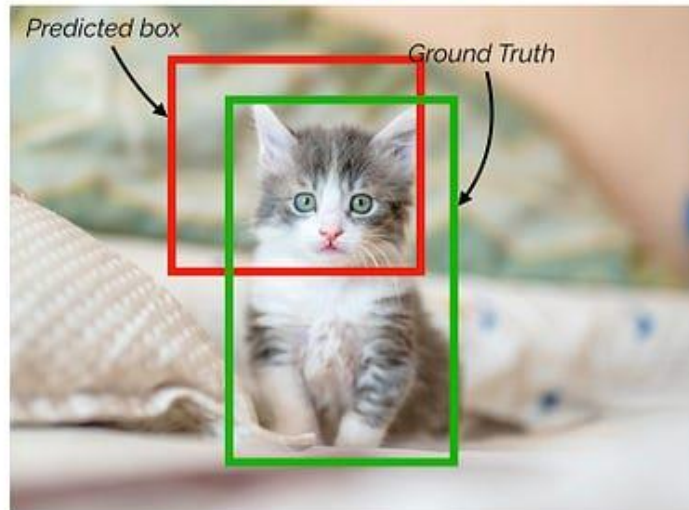


The diagram shows a single blue-filled shape representing the union of the two rectangles from the previous diagram. The top-left part of the shape is outlined in red and labeled 'Predicted box'. The bottom-right part is outlined in green and labeled 'Ground Truth'. Arrows point from the labels to their respective parts of the union shape.

Model Evaluation

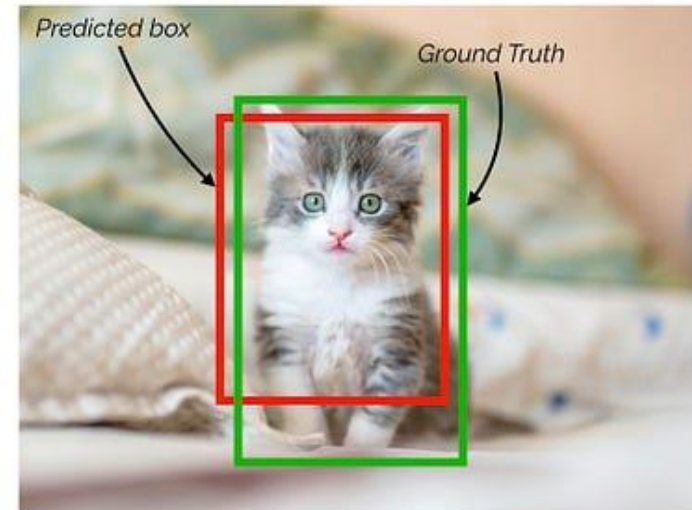
If IoU threshold = 0.5

False Positive (FP)



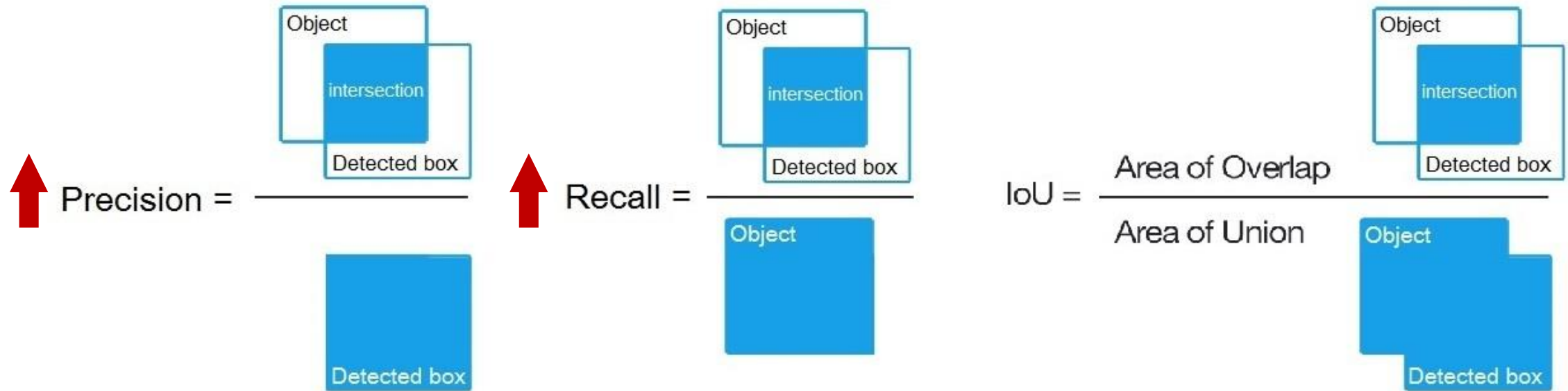
$IoU = \sim 0.3$

True Positive (TP)



$IoU = \sim 0.7$

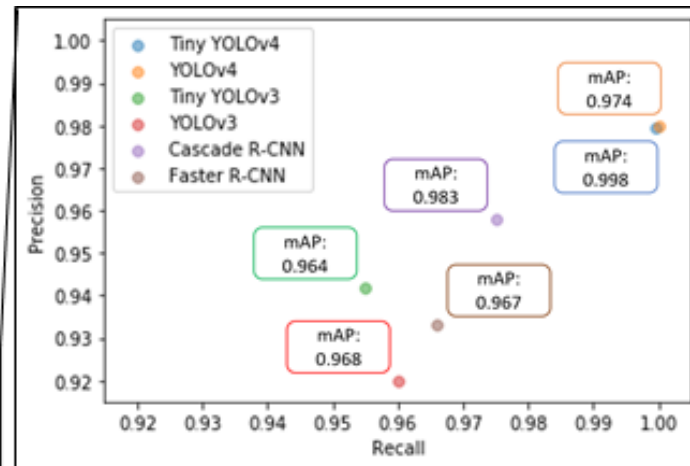
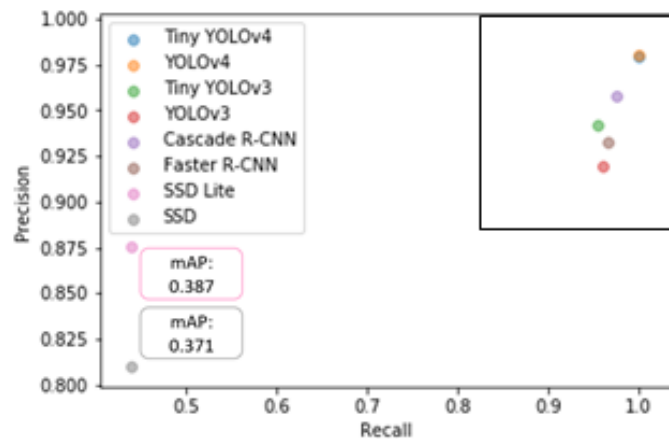
Model Evaluation



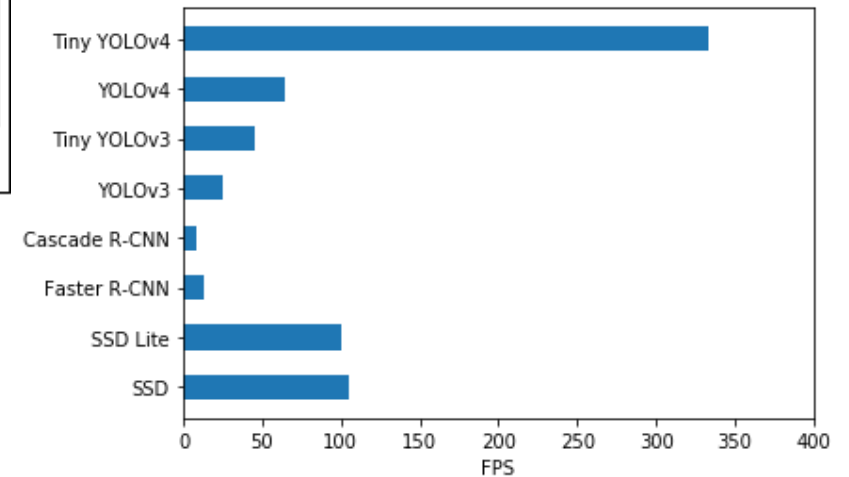
Model Performance

Real-world dataset containing single object per image and two class labels for object detection (Dataset 1)

YOLOv4 and Tiny YOLOv4 outperform other models

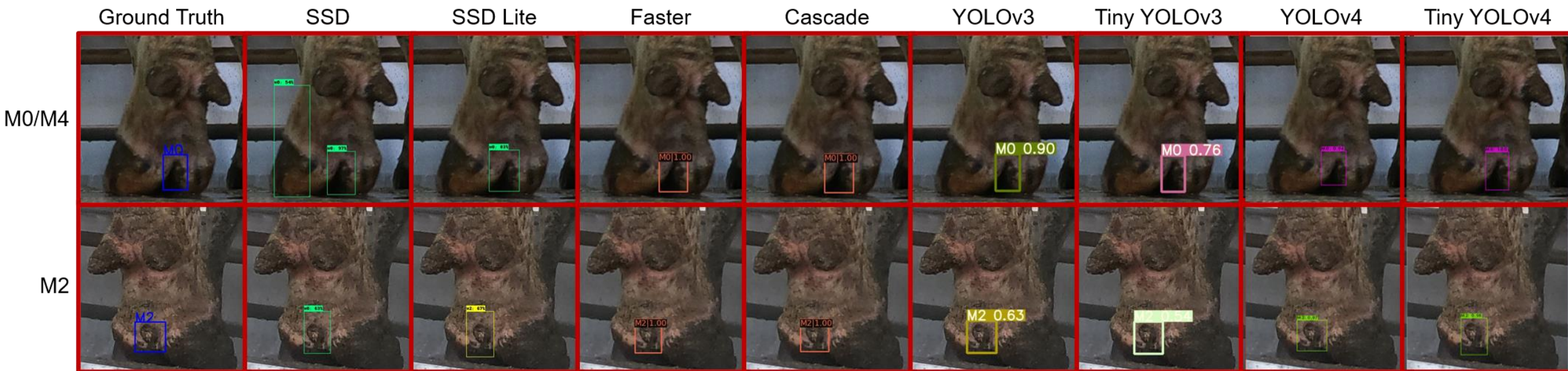


Tiny YOLOv4 is the best model for our use case

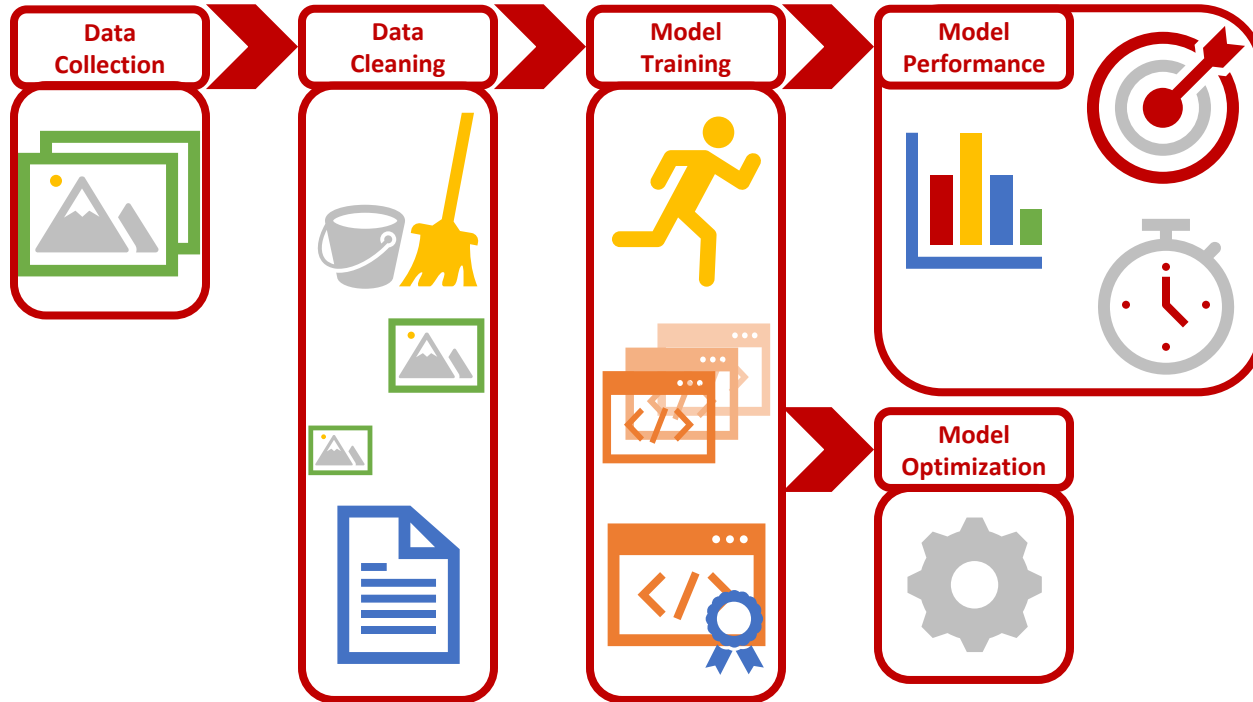


Model Performance

Real-world dataset containing single object per image and two class labels for object detection (Dataset 1)



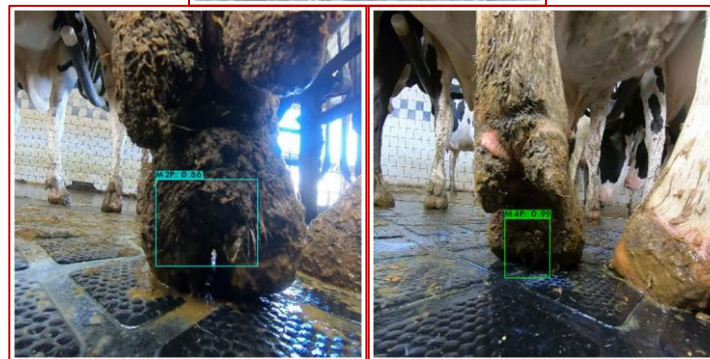
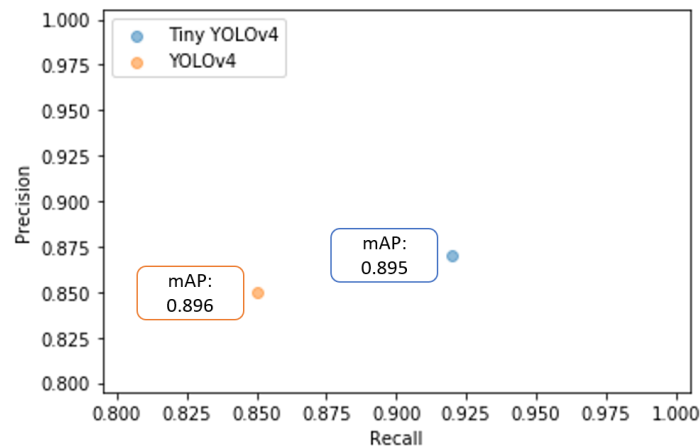
Workflow



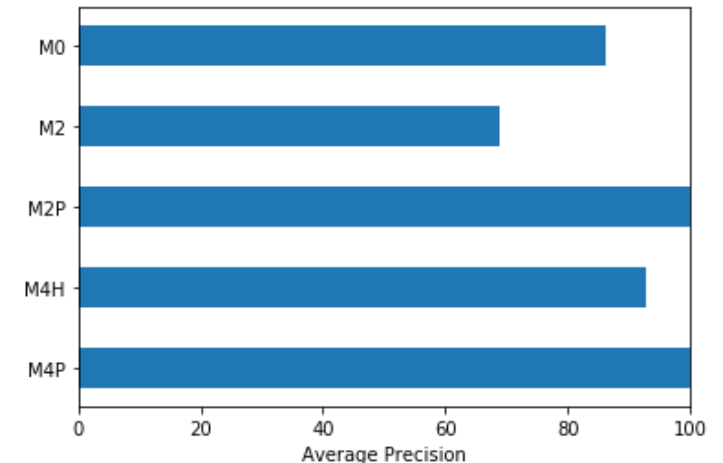
Model Optimization

Real-world dataset containing multiple objects per image and more class labels for object detection (Dataset 2)

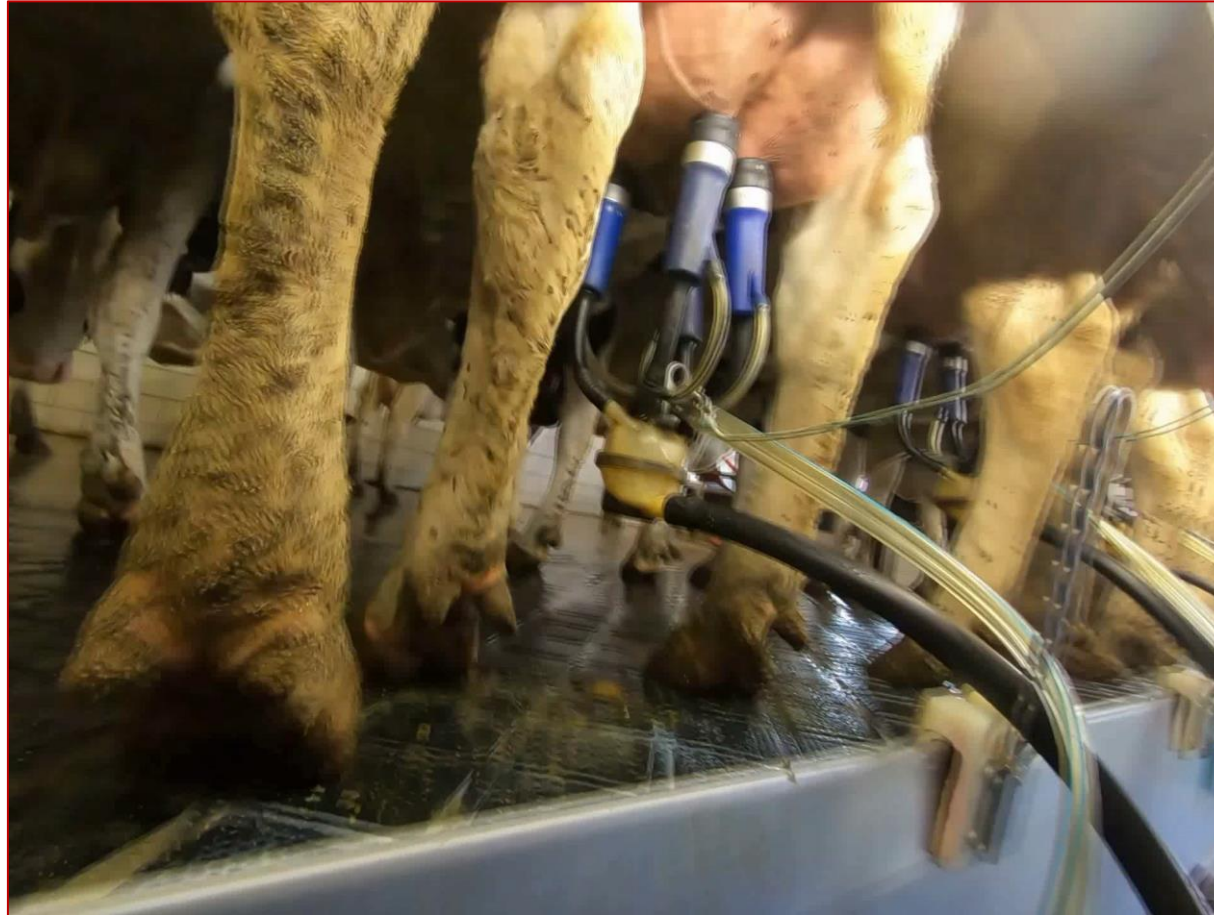
Similar performance for Tiny YOLOv4 and YOLOv4



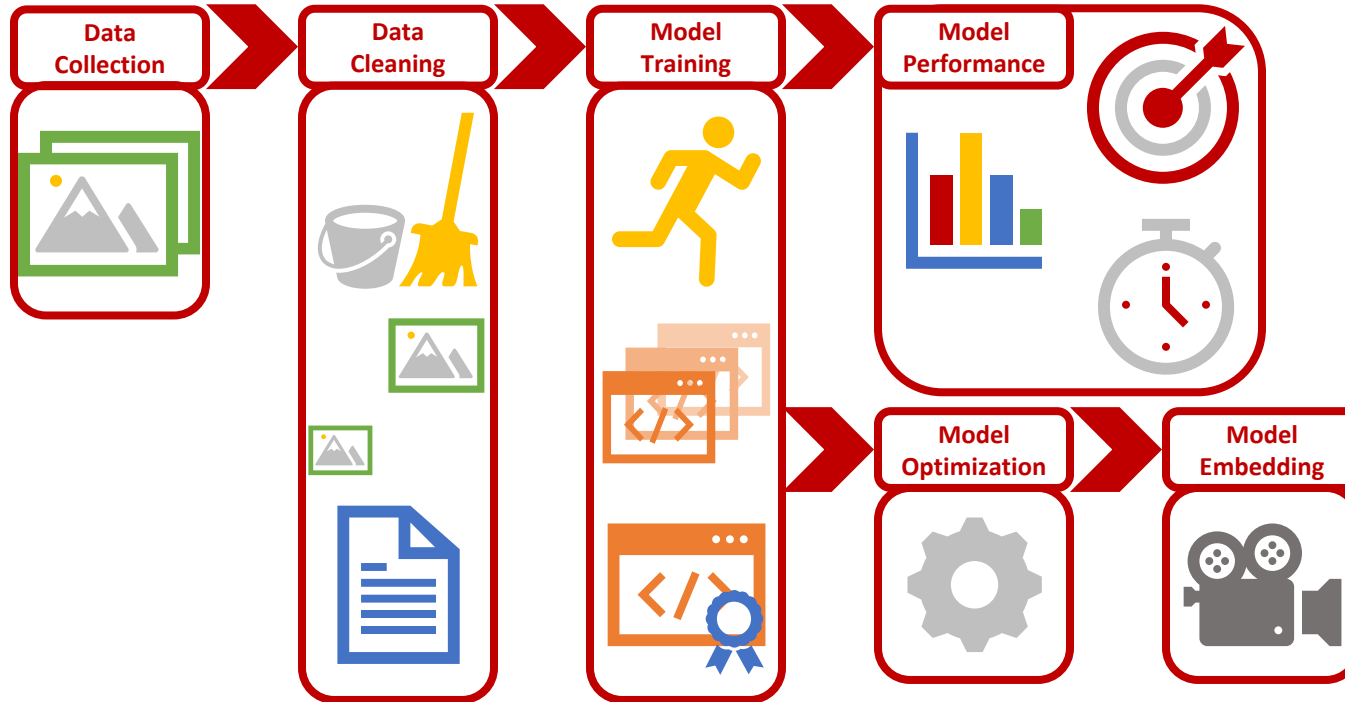
Difference in performance between classes for Tiny YOLOv4



Best Model



Workflow



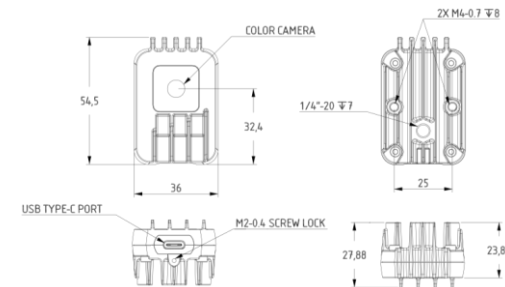
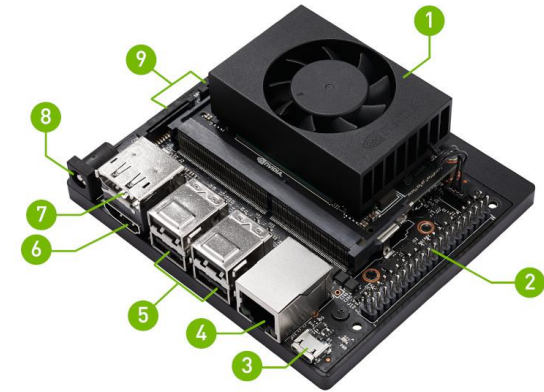
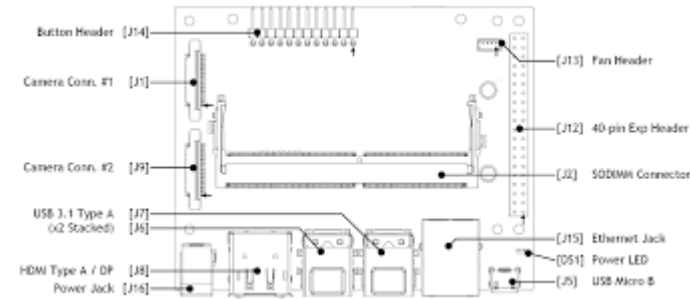
Device-Based Implementation

- Raspberry Pi
- Jetson Nano and Xavier
- OAK-1 and OAK-D Lite



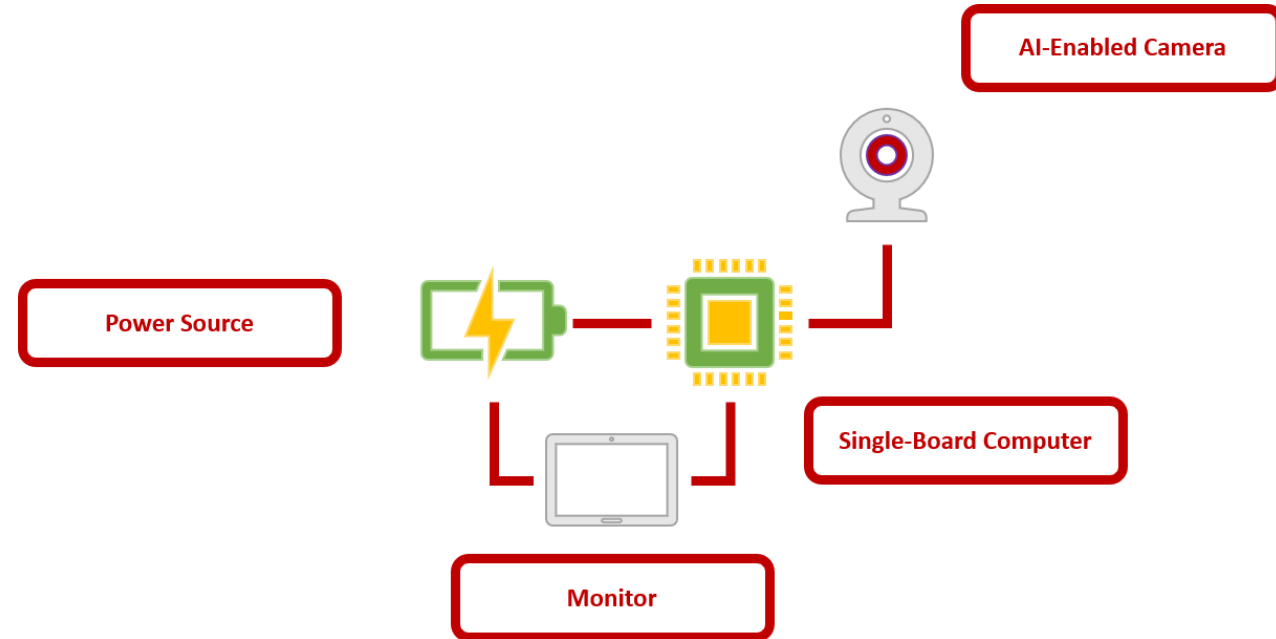
Model Embedding

- NVIDIA Jetson Xavier NX
 - Single-board computer (SBC)
 - Small, lightweight, and energy efficient
- OAK-1 camera
 - Edge accelerator
 - Intel Neural Compute Stick2
 - USB3 Type-C device power and connectivity
- LCD screen
- Power source



Model Embedding

- NVIDIA Jetson Xavier NX
 - Single-board computer (SBC)
 - Small, lightweight, and energy efficient
- OAK-1 camera
 - Edge accelerator
 - Intel Neural Compute Stick2
 - USB3 Type-C device power and connectivity
- LCD screen
- Power source



Model Embedding

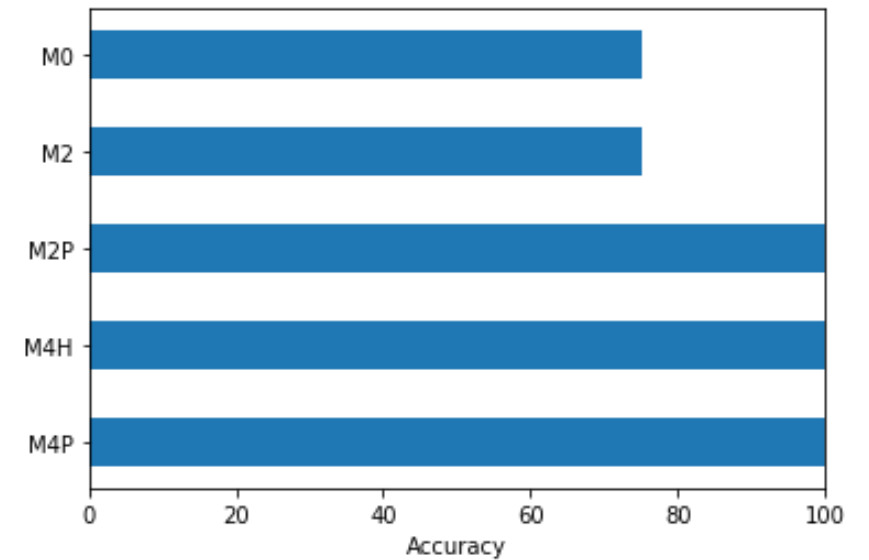
Implementation processed
images at 40 FPS



Excellent agreement with
Cohen's kappa of 0.83



Detect all five class labels on
video and webcam



Summary



SCAN ME

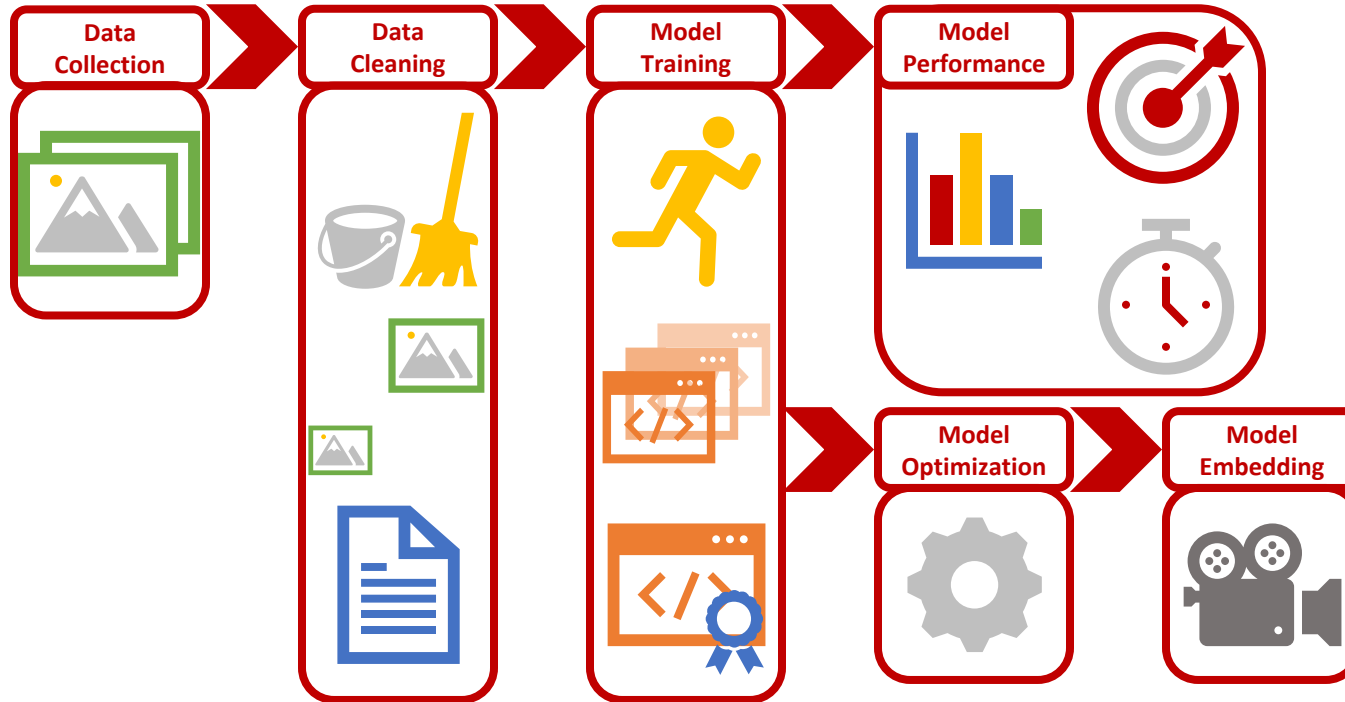
- The workflow was able to accurately and speedily detect DD on edge devices.
- YOLOv4 and Tiny YOLOv4 outperformed all other models
- Tiny YOLOv4 was the best model for our use case
- Tiny YOLOv4 on OAK-1 x NVIDIA Jetson Xavier NX edge device was fast and accurate
- Computer vision for veterinary medicine

Cloud-Based Implementation

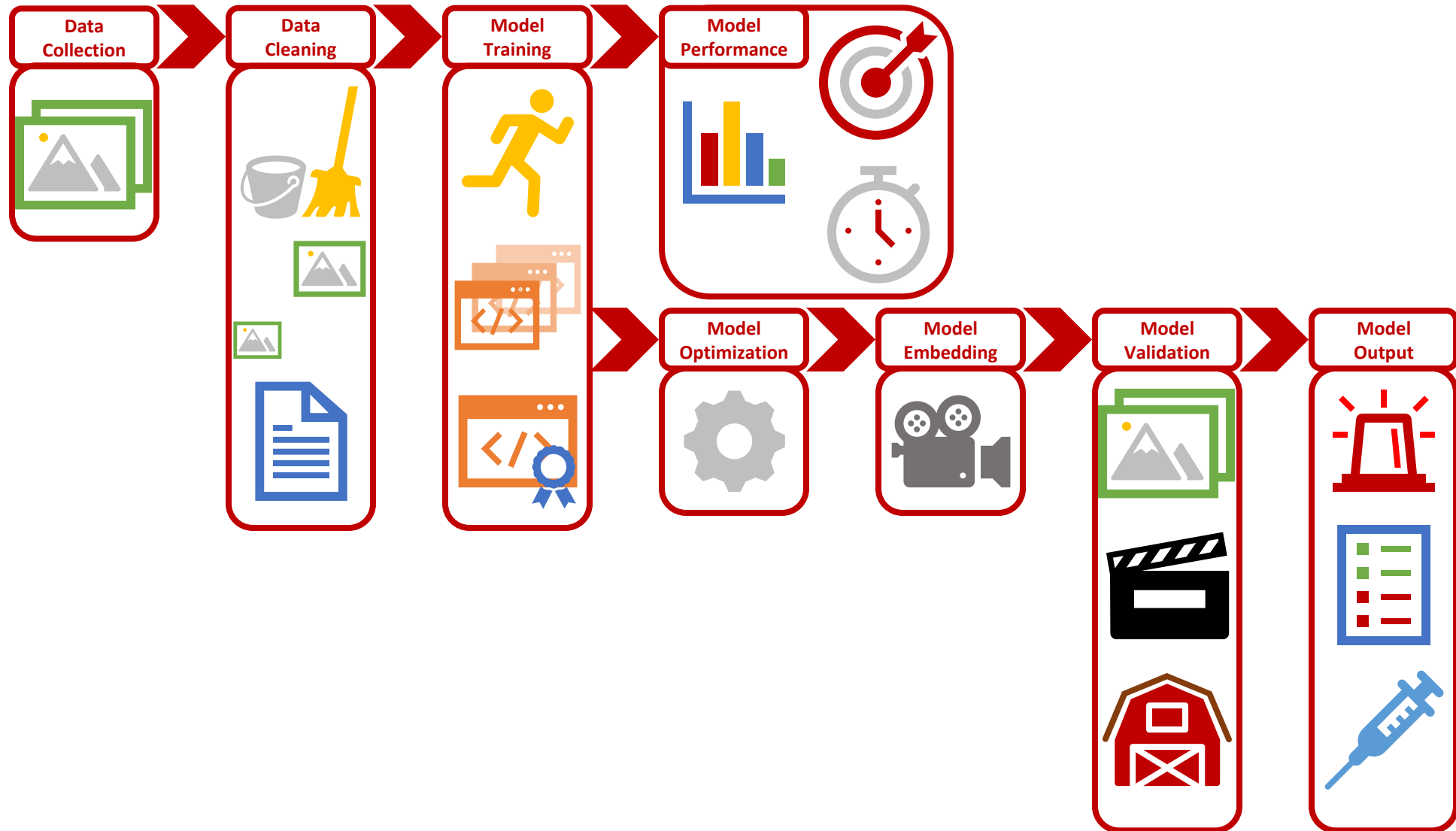
- AWS Website
- Docker Environment
- Android Application



Workflow



Summary and Future Directions



On-Going Projects

- Application using video and real-time detection
 - Device-Based Implementation
 - Cloud-Based Implementation
- Model optimization
- Model validation
- Expand to beef cattle
- Extend to other model organisms
- Other object detection algorithms
- Other programming frameworks



Acknowledgements

- Döpfer Lab (former and current)
 - Allie Hoerth
 - Juan Francisco Mandujano Reyes
 - Nathan Bollig
 - Emil Walleser
- Dr. Neoklis Apostolopoulos
- Thesis Committee
 - Prof. Guilherme Rosa
 - Prof. João Dórea
 - Prof. Paul Merkatoris
 - Prof. Brian Yandell
- US Department of Agriculture
- National Institute for Food and Agriculture - Animal Health Grant (WIS03082)



United States Department of Agriculture
National Institute of Food and Agriculture



References

- [1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142-158, 2015.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016.
- [3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21-37, Springer, 2016.
- [4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, 2017.
- [5] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154-6162, 2018.
- [6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [8] G. Jocher, "ultralytics/yolov5: v3.1 – bug, fixes and performance improvements." <https://github.com/ultralytics/yolov5>, 2020.
- [9] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 687-694, 2020.
- [10] P. Cernek, N. Bollig, K. Anklam, and D. Döpfer, "Hot topic: Detecting digital dermatitis with computer vision," *Journal of Dairy Science*, vol. 103, no. 10, pp. 9110-9115, 2020.
- [11] "Custom training — DepthAI documentation | Luxonis," Luxonis. <https://docs.luxonis.com/en/latest/pages/training/>.
- [12] "The World's Smallest AI Supercomputer," NVIDIA. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/>.
- [13] "OAK-1 — DepthAI Hardware Documentation 1.0.0 documentation," Luxonis. <https://docs.luxonis.com/projects/hardware/en/latest/pages/BW1093.html>.

Thank you!

