Submitted By

Jemish Moradiya (1417346)

Meet Kiritbhai Makadiya (1422302)

Barun Chakraborty (1381935)

Md Shahinur Rahman (1396467)

Cat-Dog Detection

Cloud Computing SoSe23

Prof. Dr. Christian Baun

Team Responsibilities

Jemish Moradiya & Meet Makadiya

- ❑ Hardware Setup
- ❑ Camera Setup
- ❑ OS install Raspberry Pi
- ❑ K3S Clusters
- ❑ Docker
- ❑ Minio
- ❑ Mqtt

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Team Responsibilities

Meet Makadiya & Jemish Moradiya

- ❑ Frontend
- ❑ Back-End
- ❑ API Integration

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Team Responsibilities

Md Shahinur Rahman & Barun Chakraborty

- ❑ Preparing Dataset
- ❑ Model Training
- ❑ Project Report
- ❑ Project Presentation

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Team Responsibilities

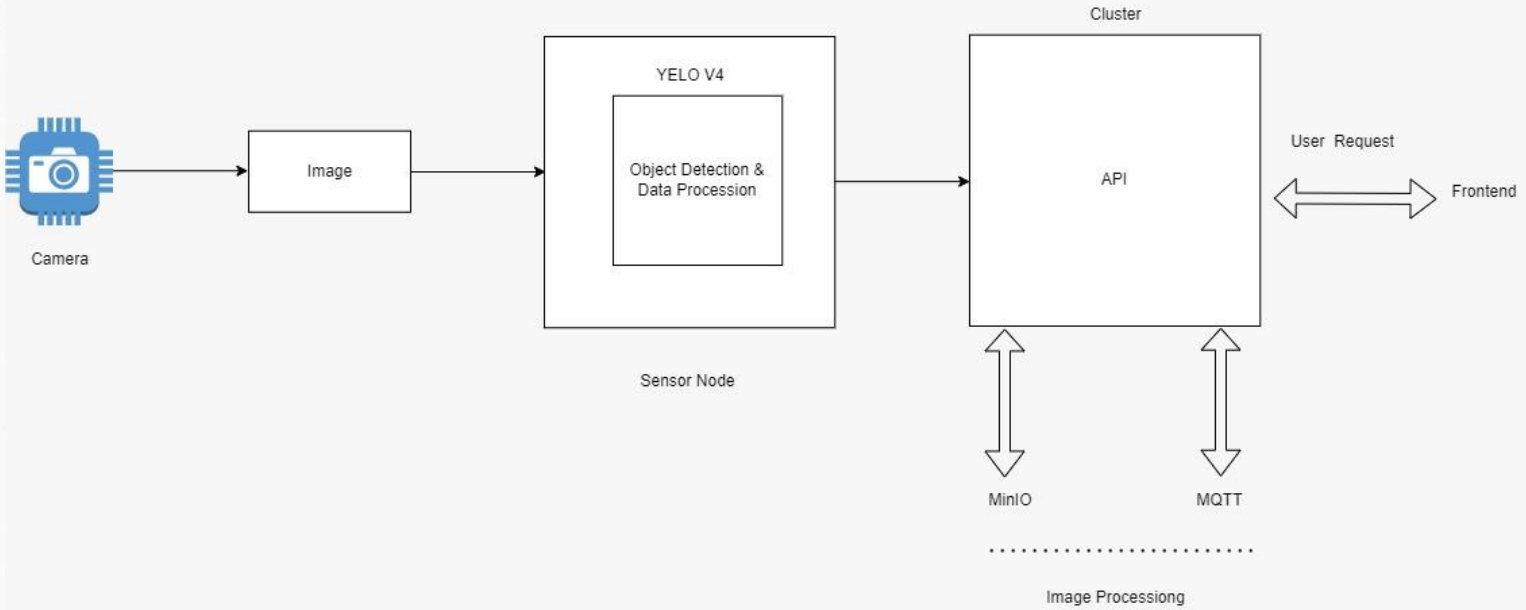
By Team Work

- ❑ Documentation

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Architecture Diagram



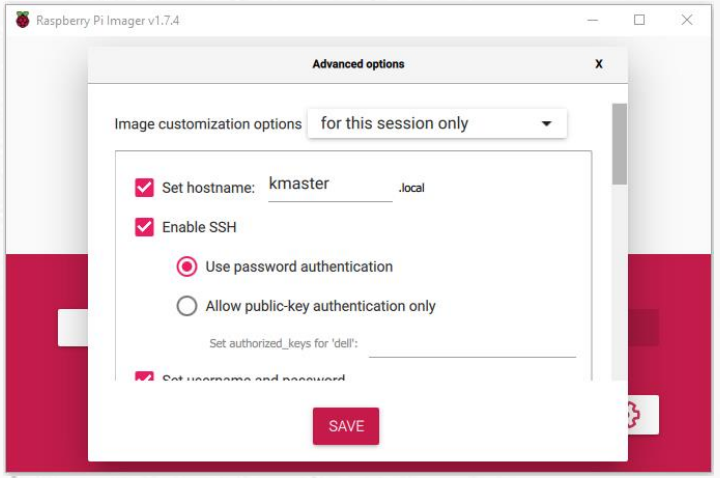
Cat-Dog Detection

Cloud Computing SoSe23

Prof. Dr. Christian Baun

Sensor/Edge Node Deployment

Setting up K3S Cluster using Raspberry Pi 3



Setting up K3S Cluster

- **Install k3s server**

- curl -sfL https://get.k3s.io | sh –

sudo cat /var/lib/rancher/k3s/server/node-token

```
pi@kmaster:~ $ sudo cat /var/lib/rancher/k3s/server/node-token
K1012630f90eb7040690d92765c5a5a8af2a082ab2e2c5f09a0d5a3cd0131677822::server:c1c7b474a2dea0a8eec0e188cf875769
```

- ***For Worker NOde***

- curl -sfL https://get.k3s.io | K3S_URL=https://<kmaster_IP_from_above>:6443 K3S_TOKEN=<token_from_above> sh –

sudo kubectl get nodes
 # or
sudo k3s kubectl get nodes

```
pi@kmaster:~ $ sudo k3s kubectl get nodes
NAME       STATUS   ROLES                  AGE   VERSION
knode2     Ready    <none>                 18h   v1.27.3+k3s1
knode1     Ready    <none>                 18h   v1.27.3+k3s1
knode3     Ready    <none>                 17h   v1.27.3+k3s1
kmaster    Ready    control-plane,master   19h   v1.27.3+k3s1
```

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Docker Setup

- ❑ Install Docker on Master Node and all three worker node using this Comma
  - sudo apt install docker
  - sudo systemctl start docker
  - sudo systemctl enable docker
  - sudo systemctl status docker

- ❑ Set up k3s server in master node
  - curl -sfL https://get.k3s.io | sh -s - --docker

- ❑ Install Docker on Worker Node and all three worker node using this Command
  - curl -sfL http://get.k3s.io | K3S_URL=http://:6443 K3S_TOKEN= sh -s - -- docker

- ❑ For checking status of docker in master node
  - sudo kubectl get nodes

```
pi@kmaster:~ $ sudo kubectl get node -o wide
NAME      STATUS   ROLES                  AGE   VERSION        INTERNAL-IP     EXTERNAL-IP   OS-IMAGE                         KERNEL-VERSION   CONTAINER-RUNTIME
knode1    Ready    <none>                 18h   v1.27.3+k3s1   192.168.0.101   <none>        Debian GNU/Linux 11 (bullseye)   6.1.21-v8+       docker://20.10.5+dfsg1
kmaster   Ready    control-plane,master   19h   v1.27.3+k3s1   192.168.0.233   <none>        Debian GNU/Linux 11 (bullseye)   6.1.21-v8+       docker://20.10.5+dfsg1
knode3    Ready    <none>                 17h   v1.27.3+k3s1   192.168.0.56    <none>        Debian GNU/Linux 11 (bullseye)   6.1.21-v8+       docker://20.10.5+dfsg1
knode2    Ready    <none>                 18h   v1.27.3+k3s1   192.168.0.67    <none>        Debian GNU/Linux 11 (bullseye)   6.1.21-v8+       docker://20.10.5+dfsg1
```

Cat-Dog Detection

Cloud Computing SoSe23

Prof. Dr. Christian Baun

After Successful deployment status

Minio Deployment

```
pi@knode3:~ $ sudo docker pull minio/minio
Using default tag: latest
latest: Pulling from minio/minio
Digest: sha256:cde7d0beaa150ec9f3323f9432c73794b43496176ae4d0bb4898625e0b7fe51b
Status: Image is up to date for minio/minio:latest
docker.io/minio/minio:latest
```

```
pi@kmaster:~ $ sudo kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS      AGE   IP           NODE     NOMINATED NODE   READINESS GATES
minio-deployment-7ddc7dffcd-8q4cc   1/1     Running   3 (11h ago)   19h   10.42.2.13   knode1   <none>           <none>
```

```
pi@kmaster:~ $ sudo kubectl get services -o wide
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP                                  PORT(S)                       AGE   SELECTOR
kubernetes      ClusterIP      10.43.0.1       <none>                                       443/TCP                       42h   <none>
minio-service   LoadBalancer   10.43.31.244    192.168.0.233,192.168.0.56,192.168.0.67      9000:32749/TCP,41117:30699/TCP   19h   app=minio
```

Mqtt Deployment

```
pi@knode2:~ $ sudo docker pull eclipse-mosquitto
Using default tag: latest
latest: Pulling from library/eclipse-mosquitto
Digest: sha256:efc3fd76a152985decdbd3768f79e4635d2e47febaeb1349d8f421a48fb0564b
Status: Image is up to date for eclipse-mosquitto:latest
docker.io/library/eclipse-mosquitto:latest
```

```
pi@kmaster:~/pestdetectionsystem/cluster_deploylemt/mosquitto $ sh deploy.sh
deployment.apps/mqtt-deployment created
configmap/mqtt-configmap created
service/mqtt-service created
```

```
pi@kmaster:~/pestdetectionsystem/cluster_deploylemt/mosquitto $ sudo kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS      AGE    IP           NODE     NOMINATED NODE   READINESS GATES
minio-deployment-7ddc7dffcd-8q4cc   1/1     Running   4 (24h ago)   2d7h   10.42.2.15   knode1   <none>           <none>
mqtt-deployment-658b7957bb-s84pp    1/1     Running   0             59s    10.42.2.16   knode1   <none>           <none>
```

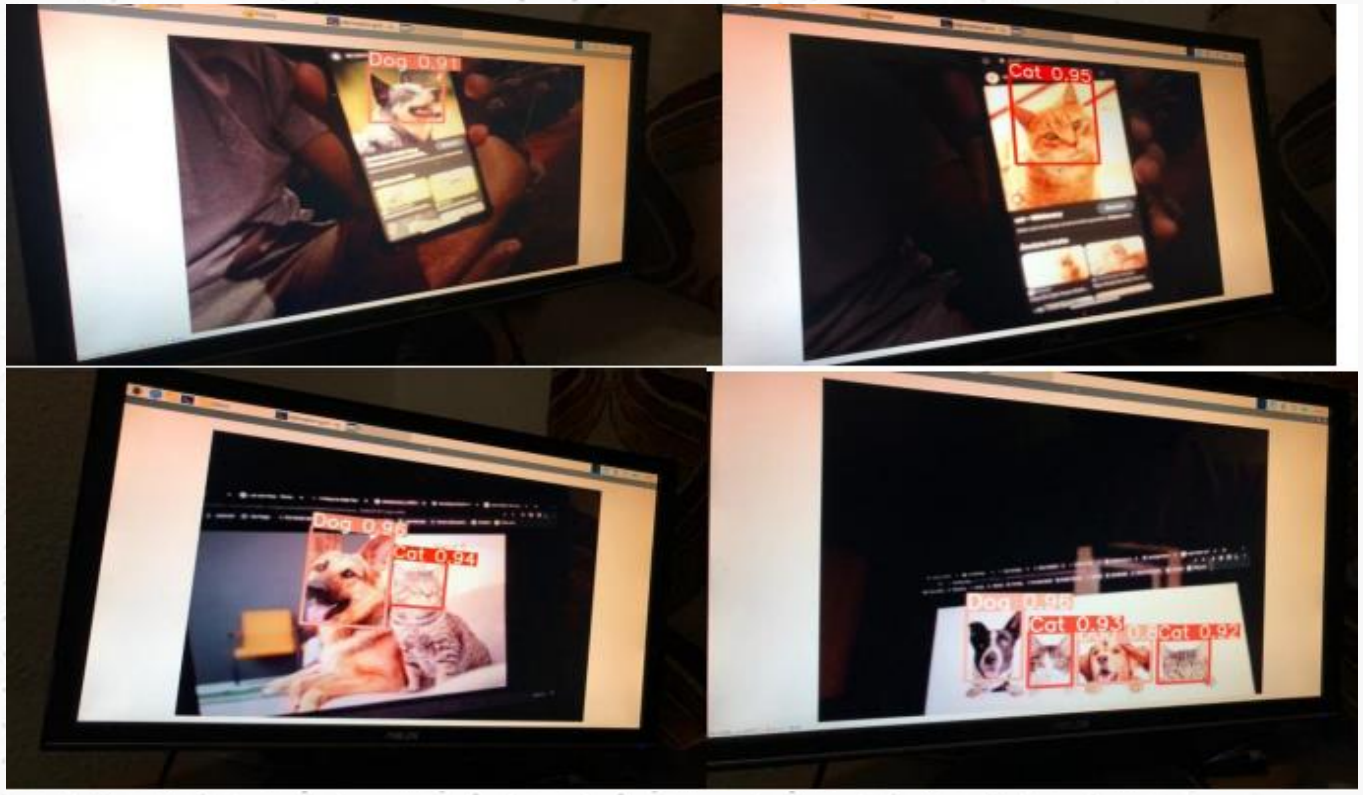
Implementation

- ❑ Camera Module and Edge Node Set Up for Object Detection
  - Connect the Pi Camera Module v2
  - OpenCV and Python Packages Setup
- ❑ Cluster Configuration
  - Docker Container Setup for master node
  - Docker Container Setup for worker node
  - Kubectl Utility Configuration
  - Deploying of MinIO in Cluster
  - Minio over Kubernetes Service Configuration
  - Deploying of MinIO in Cluster
  - MinIO with persistent volumes Activation
  - Mqtt
  - Model Training
  - Edge Node for Dog Detection
  - Upload Frame to Minio
  - Web Application Development

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Output View

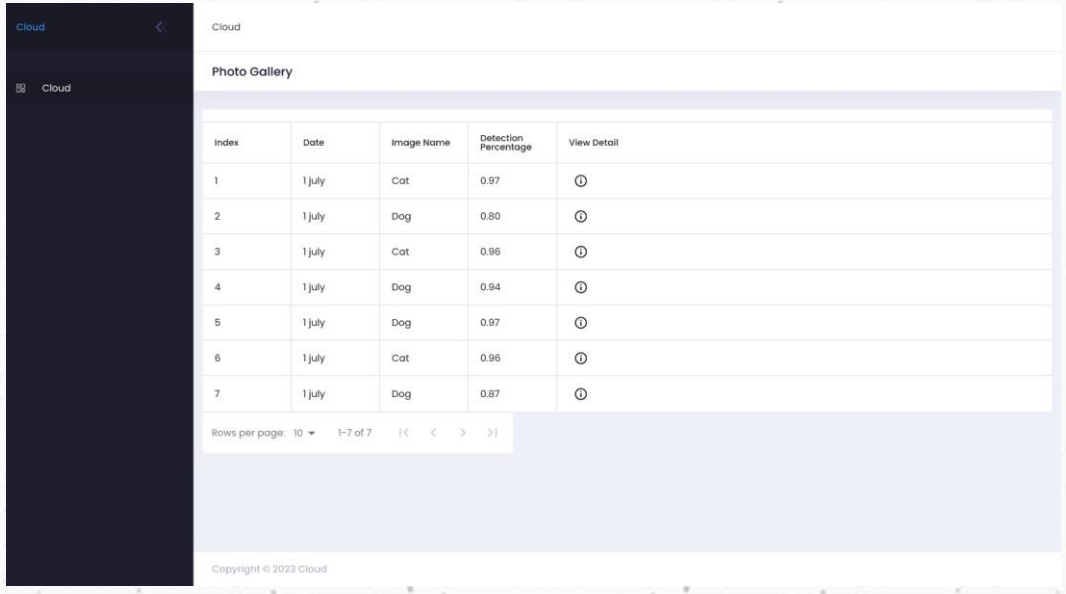


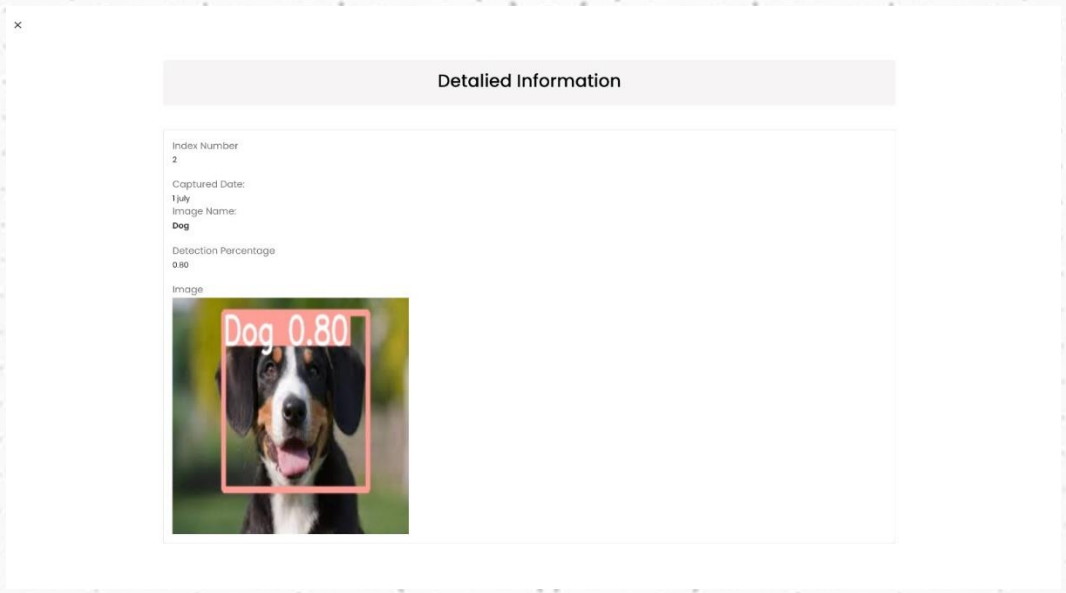
Cat-Dog Detection

Cloud Computing SoSe23

Prof. Dr. Christian Baun

Frontend View





Challenges and Improvements

- ❑Edge Node Configuration
- ❑Performance of YOLO V5 Model
- ❑Infrastructure Configuration and Sharing
- ❑Resource Constraints
- ❑Limitations of MinIO in Multi-Node Multi-Drive Mode
- ❑DHCP Server Configuration

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

References

[1] Raspberry Pi 4 Model B. Retrieved from:
https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
[2] Raspberry Pi Camera Module V2. Retrieved from:
https://www.raspberrypi.com/products/camera-module-v2/
[3] AlexeyAB/darknet GitHub Repository. Retrieved from:
https://github.com/AlexeyAB/darknet/
[4] Object Detection with OpenCV GitHub Repository. Retrieved from:
https://github.com/arunponnusamy/object-detection-opencv
[5] OpenCV Linux Installation Tutorial. Retrieved from:
https://docs.opencv.org/4.x/d7/d9f/tutorial_linux_install.html
[6] Uploading Files in MinIO Using Python. Retrieved from:
https://medium.com/featurepreneur/upload-files-in-minio-using-python-4f987f902076
[7] MinIO Linux Installation Documentation. Retrieved from:
https://min.io/docs/minio/linux/operations/installation.html

Cat-Dog Detection

Cloud Computing SoSe23
Prof. Dr. Christian Baun

Thank you

Cat-Dog Detection

Cloud Computing SoSe23

Prof. Dr. Christian Baun