2022

# Deep Learning & Food Storage

FoodiX - Brasil'IA

# Famine, food storage and AI

- Famine: one death every 4 seconds
- 40% of global food waste are due to stockage problems
- FoodiX: Connected silos company
- **Idea**: optimize placement of silos to reduce waste (e.g. Transportation)
- **Our solution**: a 95%-efficient AI silo detector from satellite images

# Outline

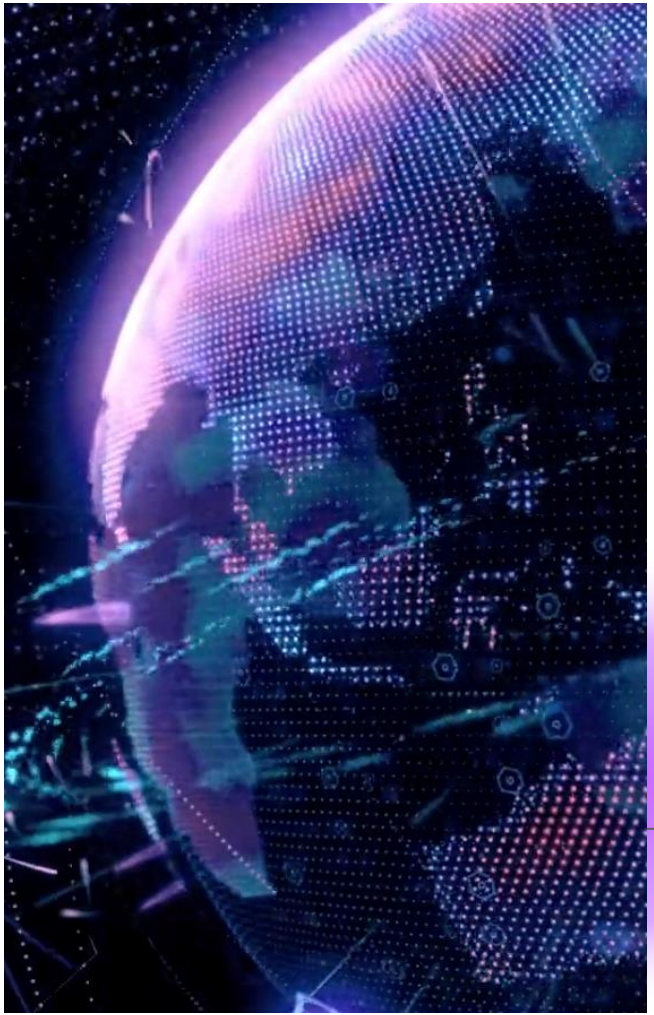
## 01 Classification & segmentation

Robust deep models for silos detection

## 02 Web App

Interactive, user-friendly and robust

## 03 Project impact





# 01

# Classification & segmentation

Robust deep models for silos detection

# Model choice classification

CNN #1

Simple model

First paper

ResNet

CNN #2

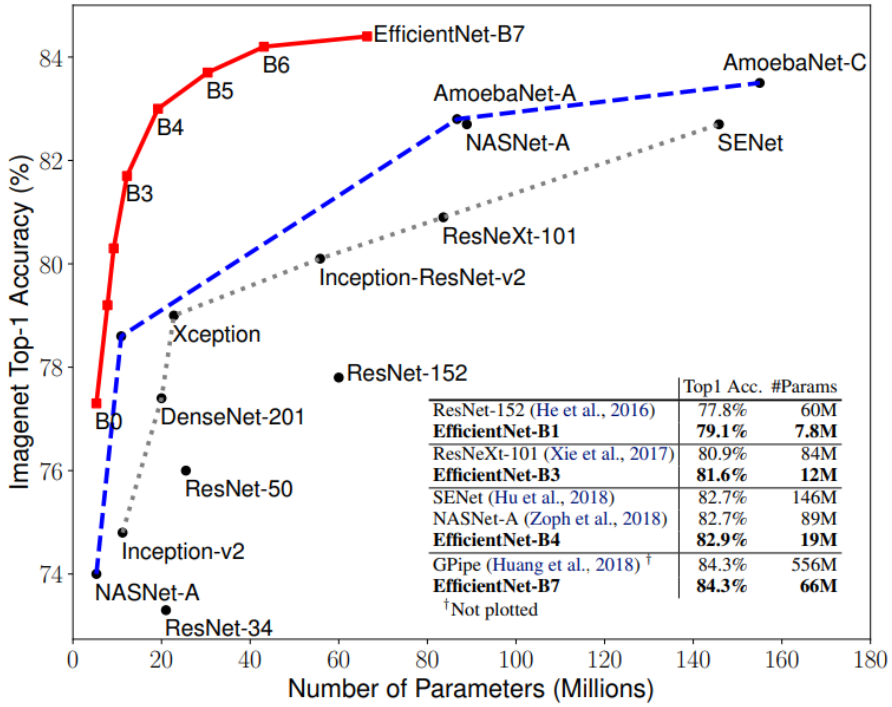
Continuation of CNN #1

OUR CHOICE

EfficientNet

# Model choice classification

## EfficientNet




| | Top1 Acc. | #Params |
|---|---|---|
| ResNet-152 (He et al., 2016) | 77.8% | 60M |
| **EfficientNet-B1** | **79.1%** | **7.8M** |
| ResNeXt-101 (Xie et al., 2017) | 80.9% | 84M |
| **EfficientNet-B3** | **81.6%** | **12M** |
| SENet (Hu et al., 2018) | 82.7% | 146M |
| NASNet-A (Zoph et al., 2018) | 82.7% | 89M |
| **EfficientNet-B4** | **82.9%** | **19M** |
| GPipe (Huang et al., 2018) † | 84.3% | 556M |
| **EfficientNet-B7** | **84.3%** | **66M** |

†Not plotted

# Why EfficientNet?

## Performance

Max achieved efficiency:

**95%**

## Deep model

Avoids overfitting from big linear layers

**18**

layers

## # parameters

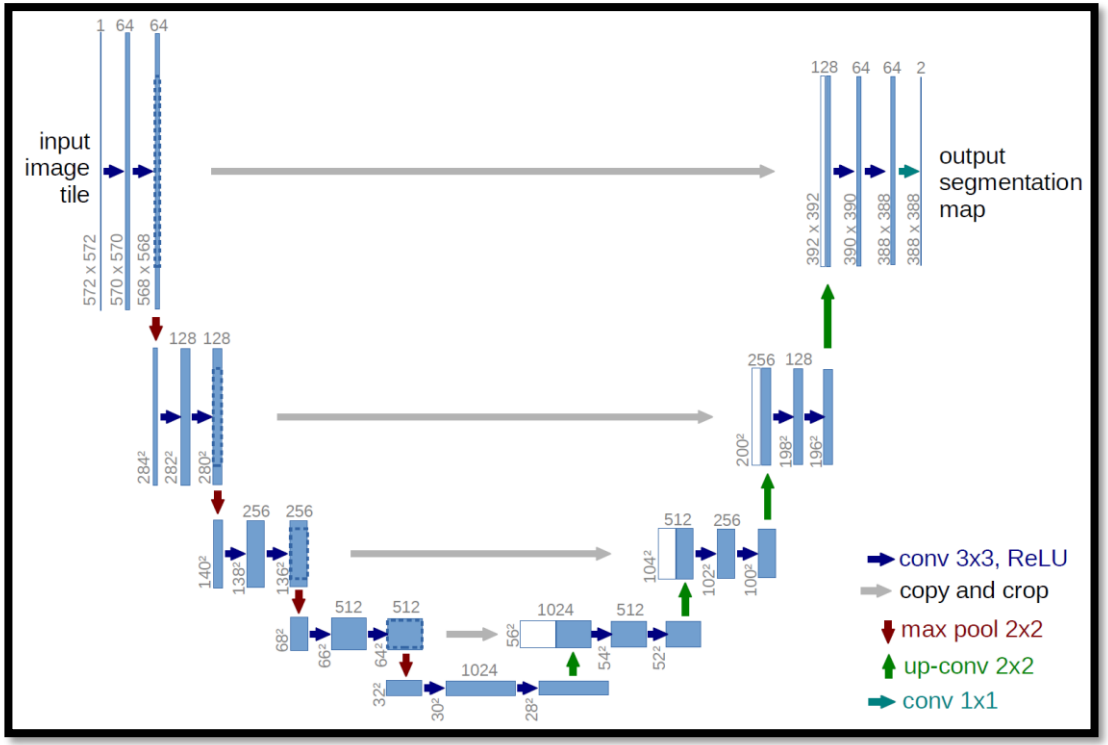
Smaller than other compared models

**3.2M**

params

# Model choice segmentation

## UNet



# Anti-overfitting procedures

## Dropout

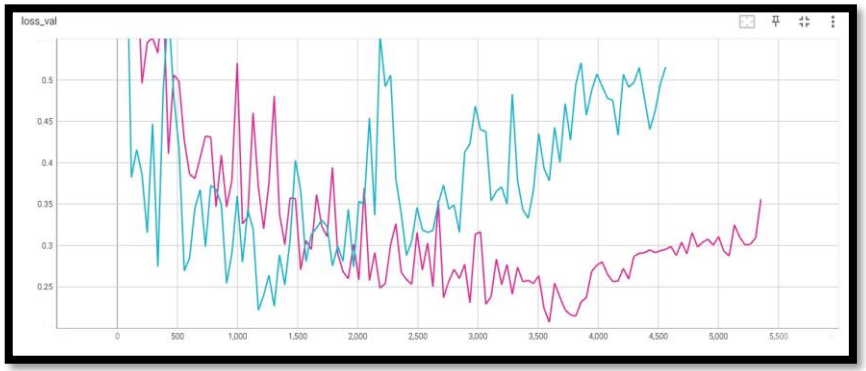
## Data augmentation

Lack os images in dataset

## Batch normalization

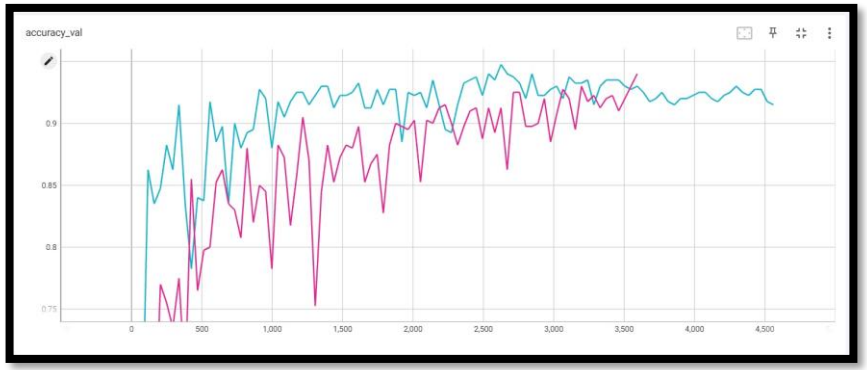
Further development: ensemble methods

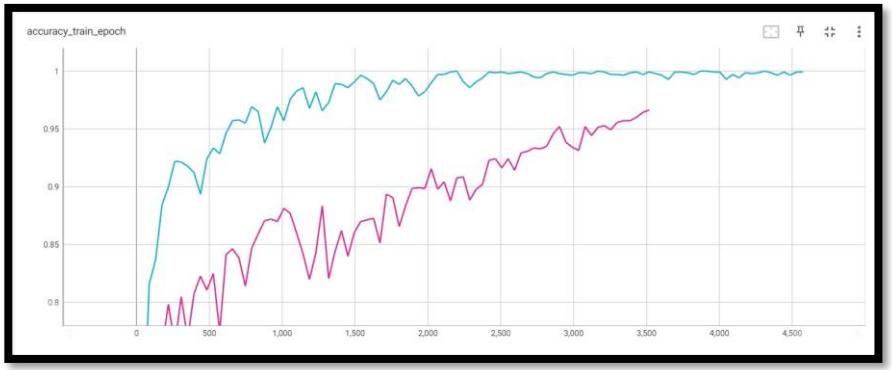
# Dropout // Anti-overfitting procedures




Dropout ⇒ decreasing loss trend

Improvement in longer time windows



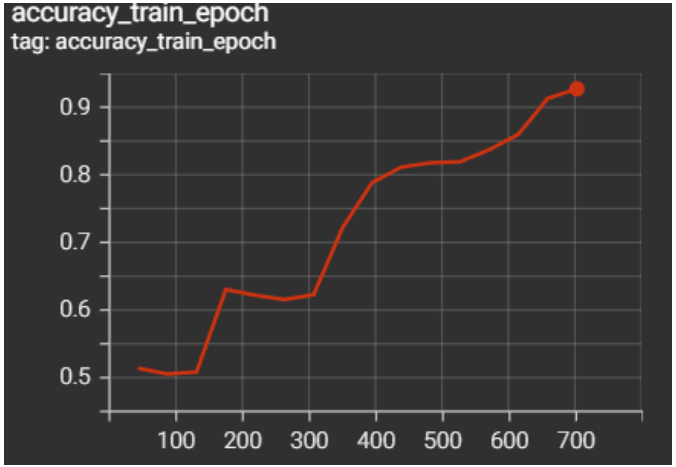



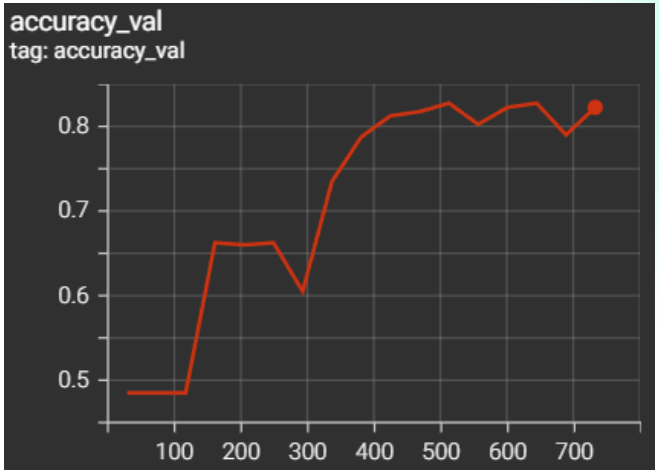
# Data augmentation

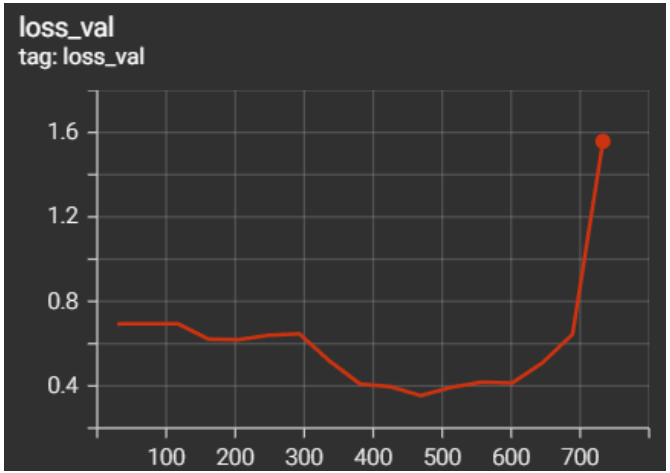
// Anti-overfitting procedures

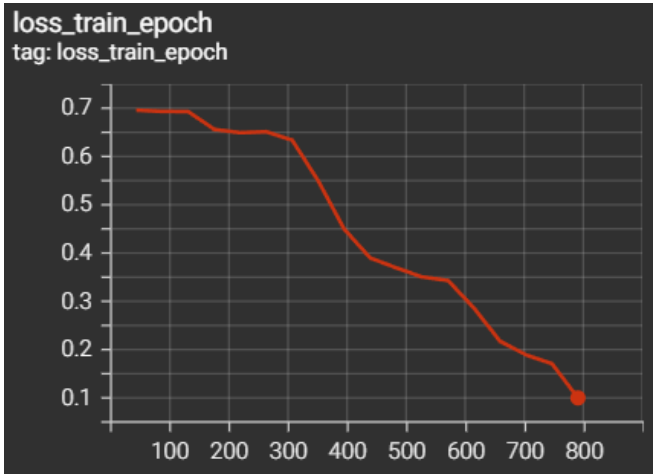
# Model performances

# CNN #1

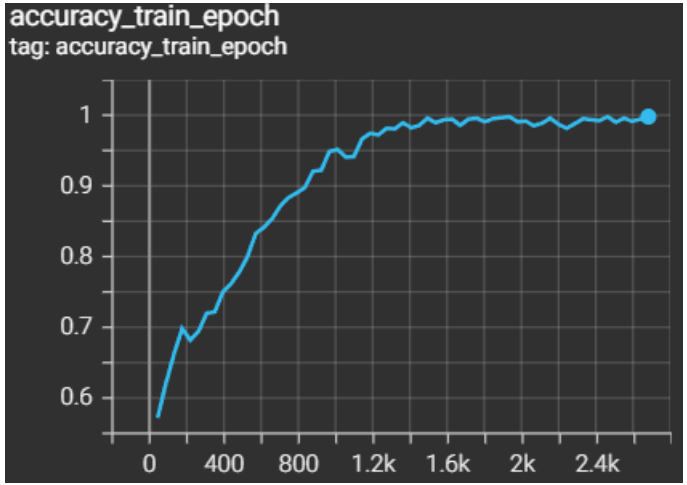

accuracy_train_epoch
tag: accuracy_train_epoch
0.9
0.8
0.7
0.6
0.5
100
200
300
400
500
600
700

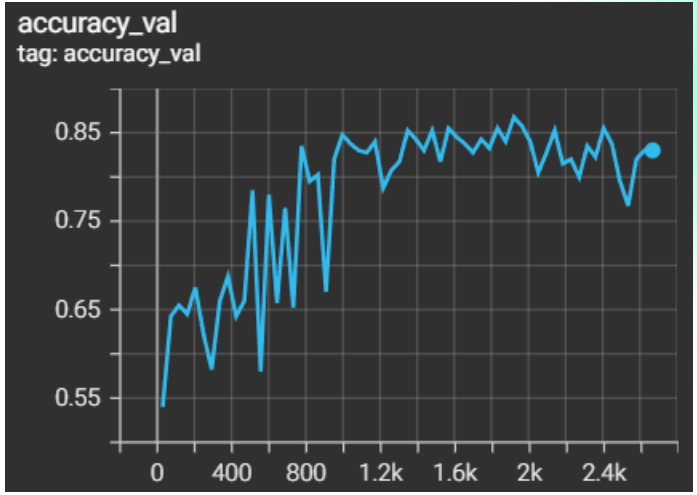

accuracy_val
tag: accuracy_val
0.8
0.7
0.6
0.5
100
200
300
400
500
600
700

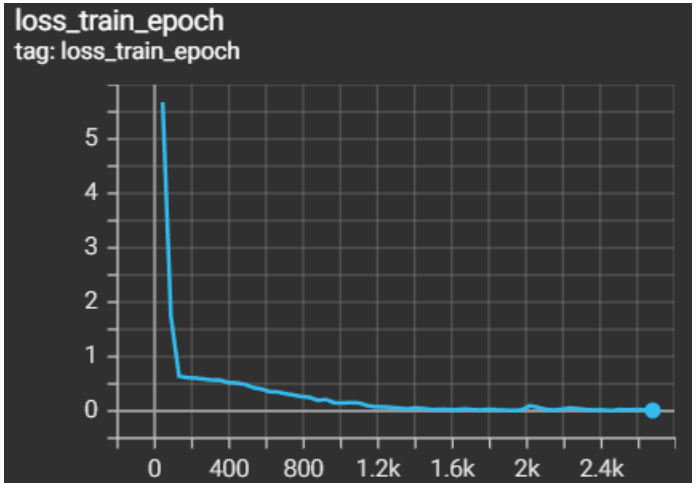

loss_val
tag: loss_val
1.6
1.2
0.8
0.4
100
200
300
400
500
600
700

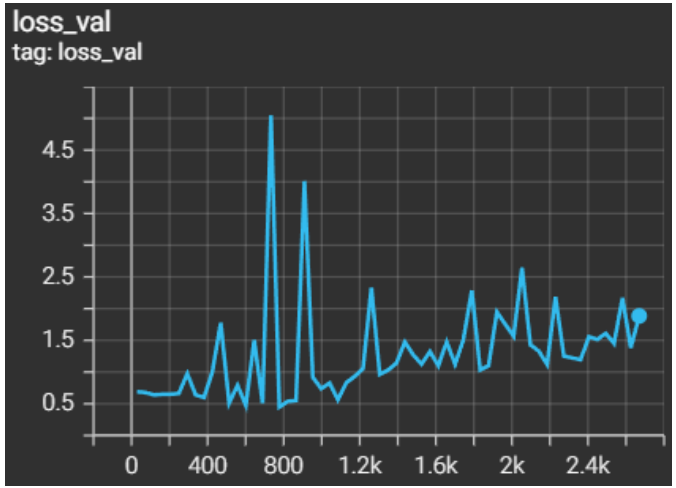

loss_train_epoch
tag: loss_train_epoch
0.7
0.6
0.5
0.4
0.3
0.2
0.1
100
200
300
400
500
600
700
800

# ResNet

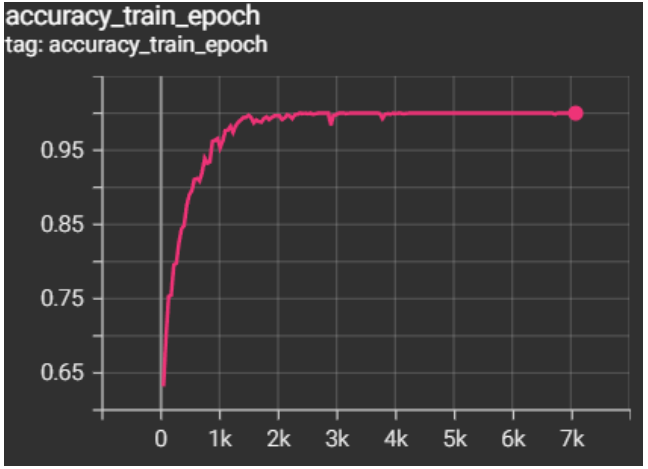

accuracy_train_epoch
tag: accuracy_train_epoch
1
0.9
0.8
0.7
0.6
0
400
800
1.2k
1.6k
2k
2.4k

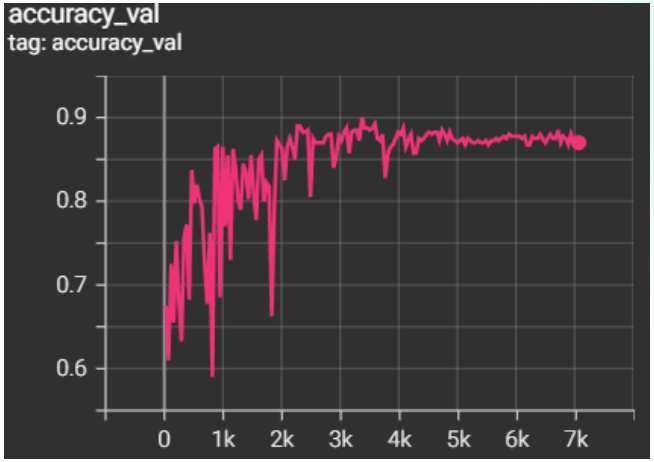

accuracy_val
tag: accuracy_val
0.85
0.75
0.65
0.55
0
400
800
1.2k
1.6k
2k
2.4k

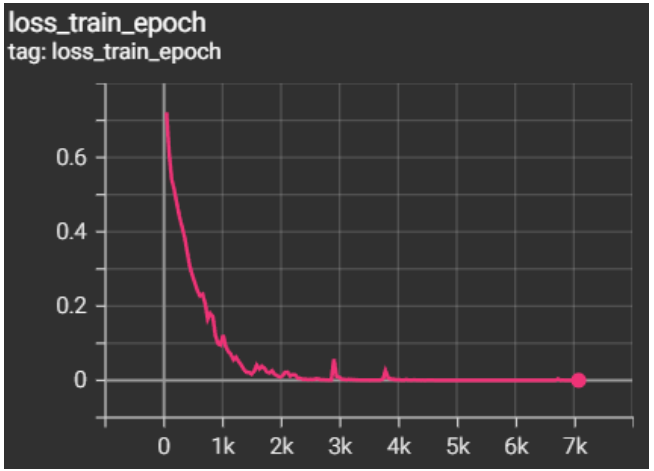

loss_train_epoch
tag: loss_train_epoch
5
4
3
2
1
0
0
400
800
1.2k
1.6k
2k
2.4k

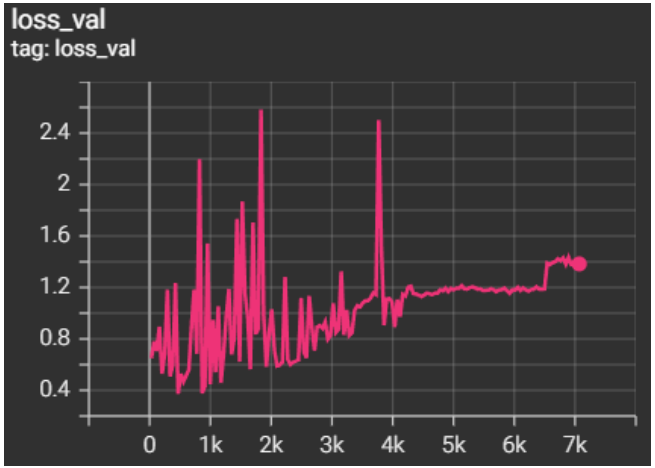

loss_val
tag: loss_val
4.5
3.5
2.5
1.5
0.5
0
400
800
1.2k
1.6k
2k
2.4k

# CNN #2

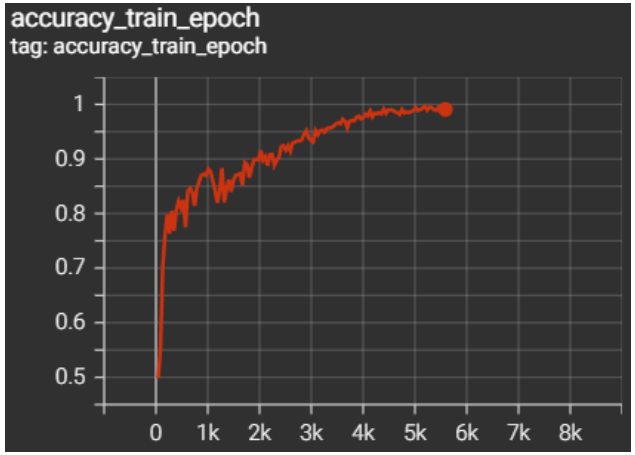

accuracy_train_epoch
tag: accuracy_train_epoch
0.95
0.85
0.75
0.65
0
1k
2k
3k
4k
5k
6k
7k

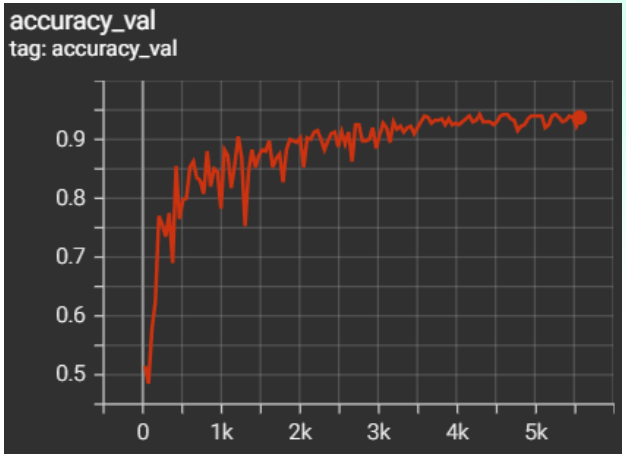

accuracy_val
tag: accuracy_val
0.9
0.8
0.7
0.6
0
1k
2k
3k
4k
5k
6k
7k

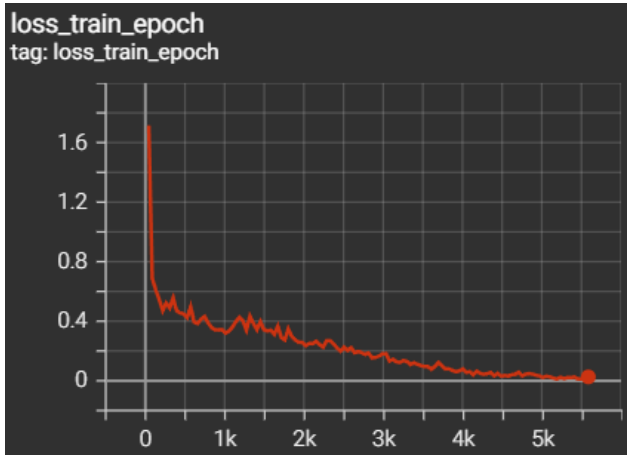

loss_train_epoch
tag: loss_train_epoch
0.6
0.4
0.2
0
0
1k
2k
3k
4k
5k
6k
7k

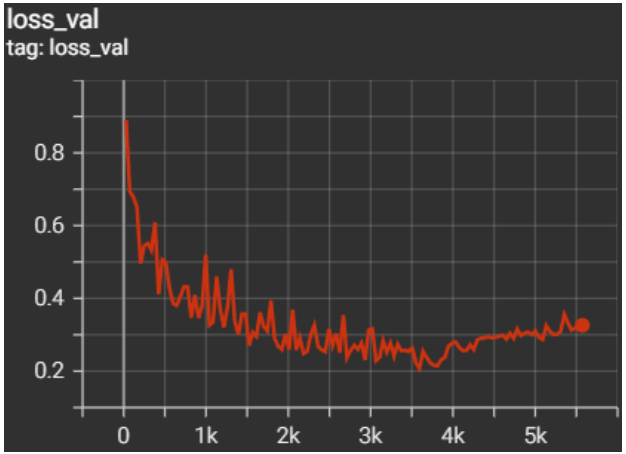

loss_val
tag: loss_val
2.4
2
1.6
1.2
0.8
0.4
0
1k
2k
3k
4k
5k
6k
7k

# EfficientNet



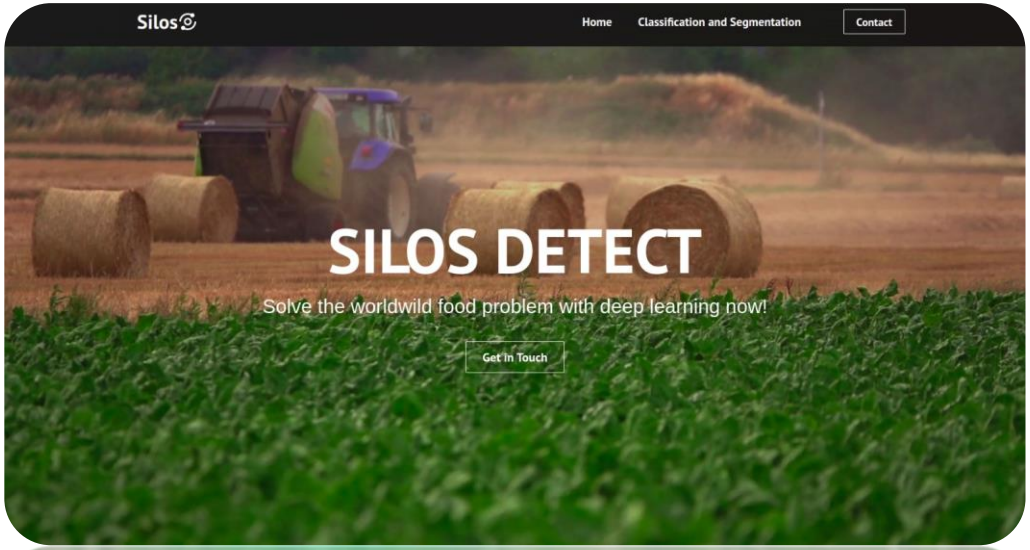



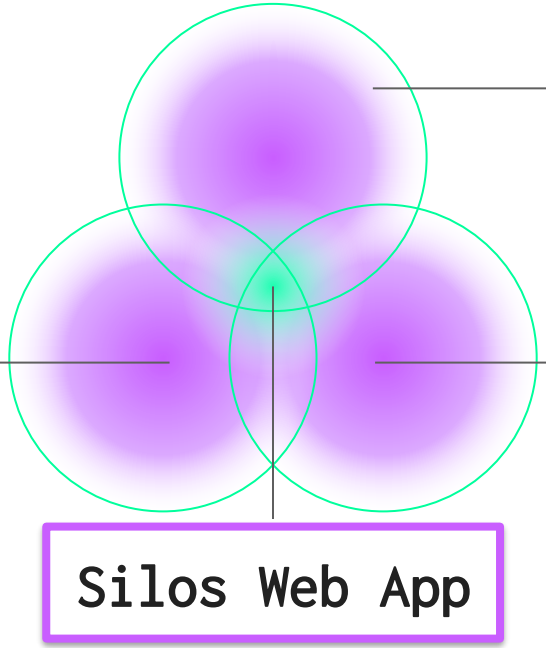



2022

# 02 Web App



# Silos Detect Web App




## User experience

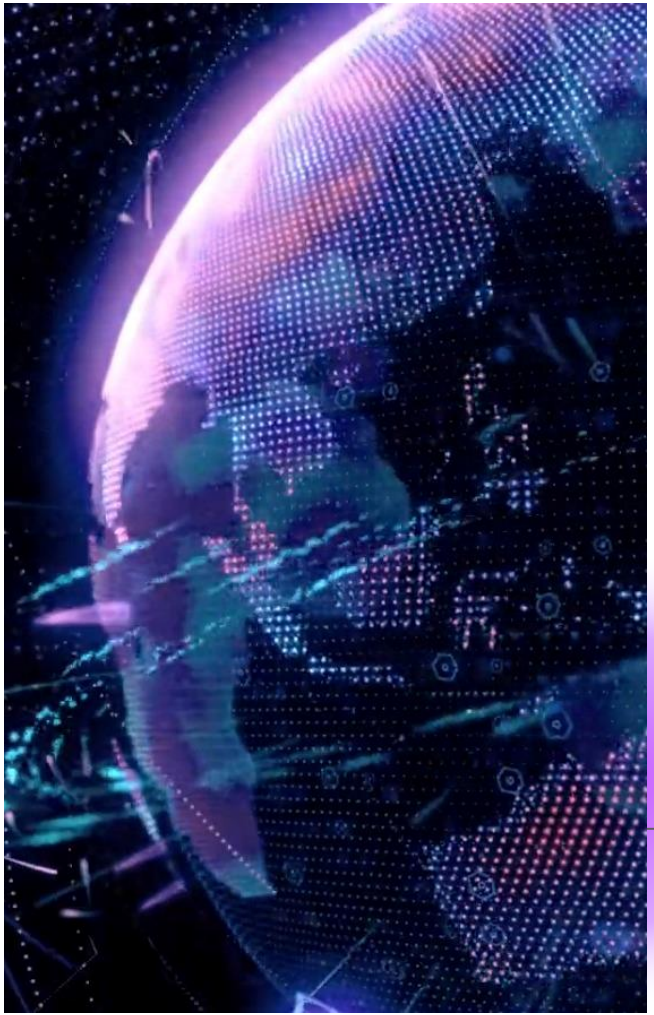
Get predictions from input images with StreamLit

## React-based

Allows iteractive and modern site

## Design

Objective and minimalist



# 03

# Product impact

Robust deep models for silos detection

# Product impact

- Use of state-of-the-art AI deep models
- High efficiency
- Flexibility to detect other buildings / elements
- Use case #1: construct heatmaps of silos
- Use case #2: data analysis with climate and positioning

# End of Tech presentation