Real-Time Weather Classifier on Edge Devices (KV260)

Group 13

黃羿寧 109511209

林芮羽 109511208

林錦樑 109511219

Outline

1 Introduction

- Our Project
- Related Work
- Workflow

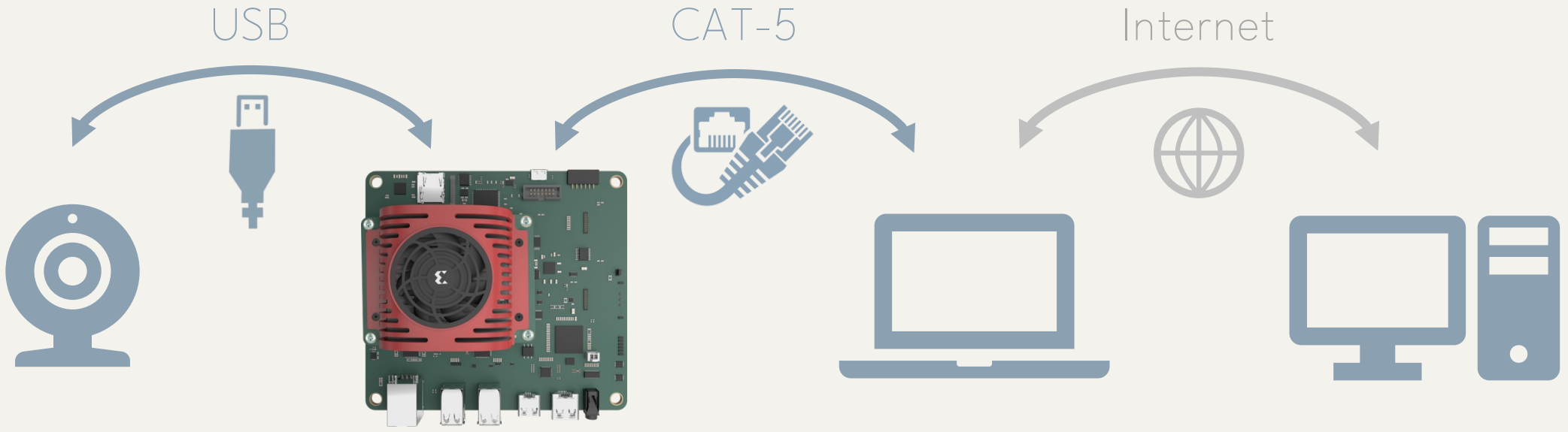
2 Implementation

- Related Works
- Dataset & Model
- Deployment

3 Conclusion

- Result (Demo)
- Obstacles
- Improvements

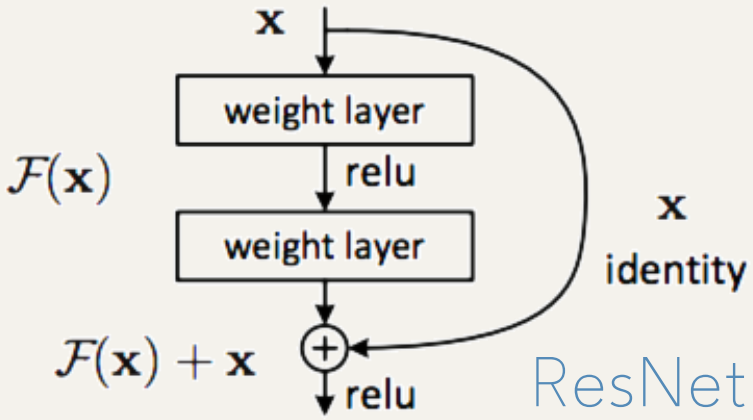
Real-Time Weather Classifier



Webcam	KV260	PC	Workstation
Capture the image	Classify the image	Show the result Train the model	X Optimize the model

Related Works

PyTorch

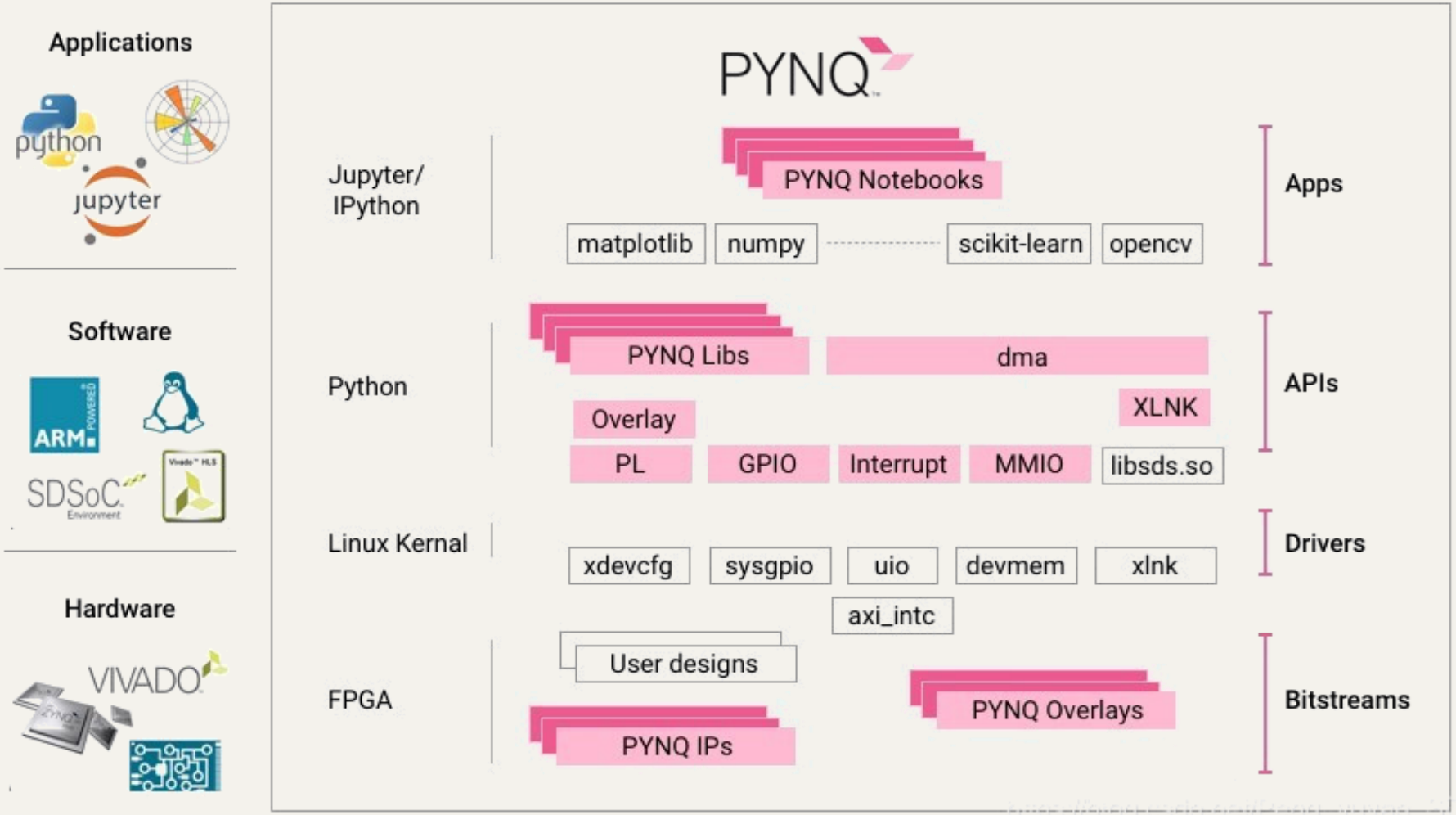


ResNet

XILINX VITIS™ | AI

OpenCV

Related Works



Workflow

Train & save the model
(PC)

- Train the model with PyTorch and save the model as .pth file.

▼

Inspect, Quantize & Compile
(Workstation)

- After inspecting, quantizing, and compiling, we get an .xmodel file.

▼

Design the main function
(KV260)

- Use DPU overlay to load the file for prediction.
- Read the images and generate the result.

Image2Weather Dataset

Weather	Number
Sunny	70,501
Cloudy	45,662
Snowy	1,252
Rainy	1,369
Foggy	357
Other	64,657
Total	117,532

*Used in training

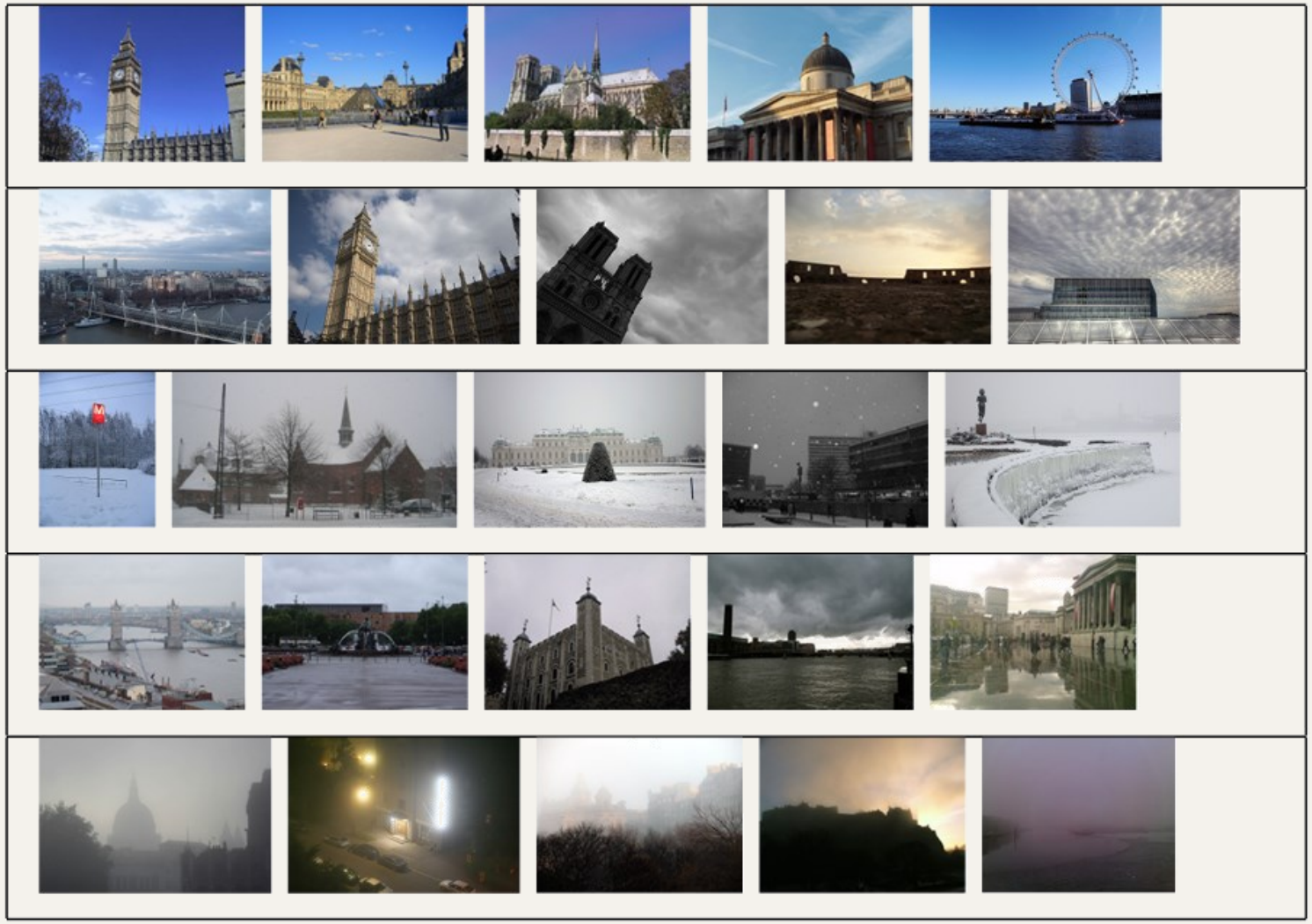
Image2Weather Dataset

Sunny

Cloudy

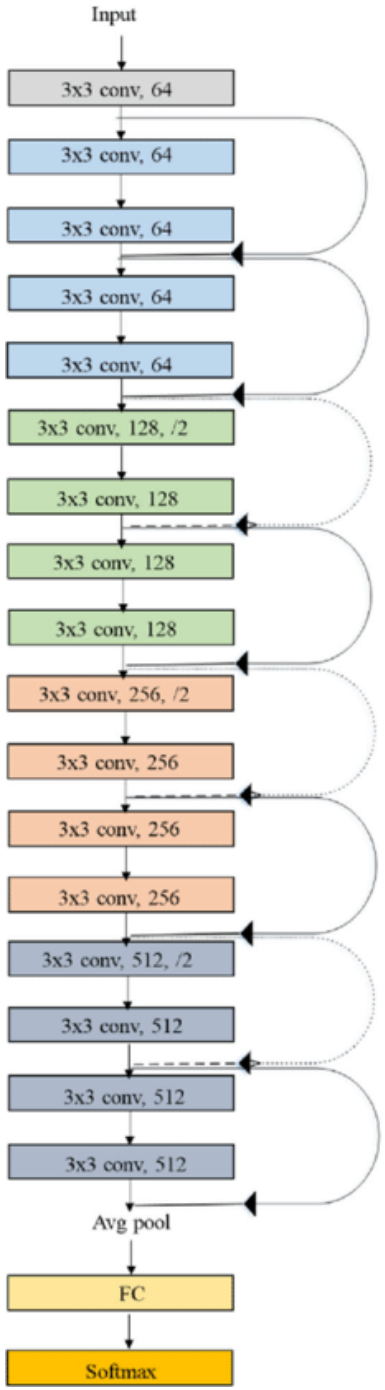
Rainy

Image2Weather Dataset



ResNet18

```python
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.resnet18 = models.resnet18(weights=models.ResNet18_Weights.IMAGENET1K_V1)
        self.fc = nn.Linear(self.resnet18.fc.in_features, 3)
        self.resnet18.fc = self.fc

    def forward(self, x):
        x = self.resnet18(x)
        return x
```



https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248

Hyperparameters

- Batch size = 64
- Epoch = 15
- Criterion = CrossEntropyLoss
- Optimizer = Adam
- Learning Rate = 0.001
- Scheduler = StepLR(optimizer, step_size=5, gamma=0.5)

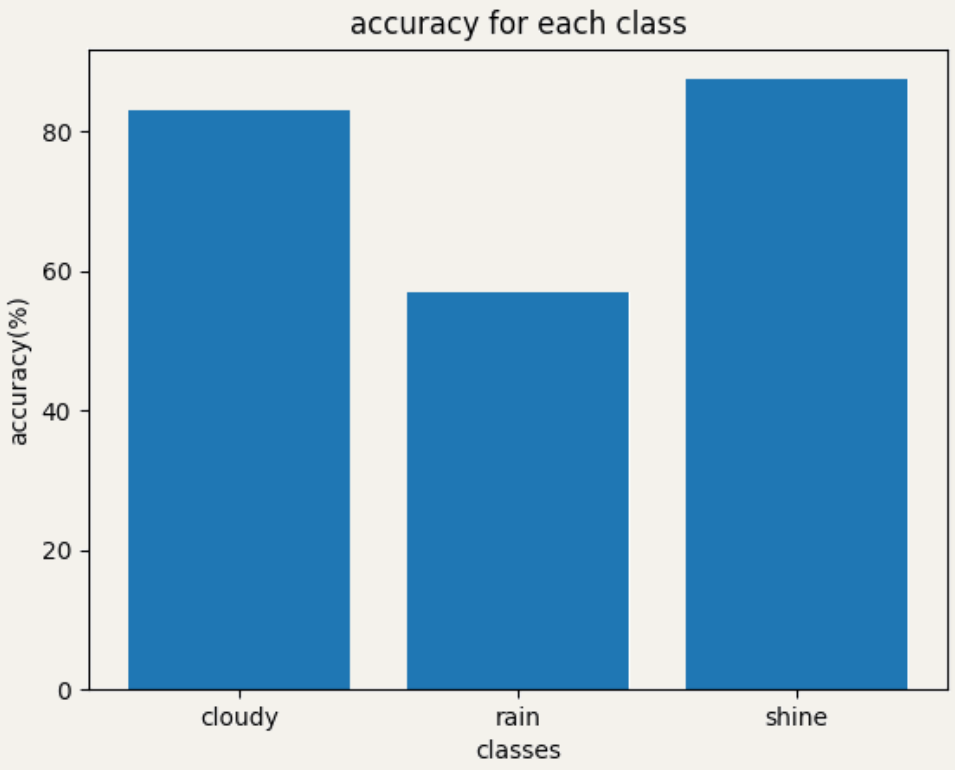
Imbalanced data

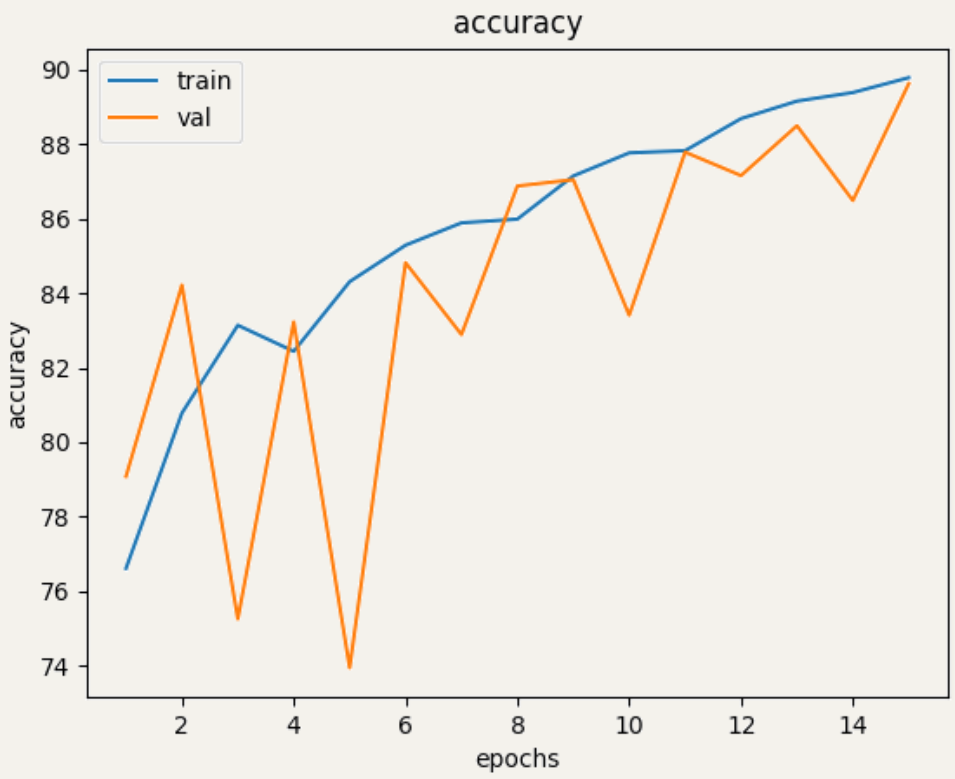
☃ Re-Weighting

- ☂ class_weights: $weight_{class} = \frac{\sum sample_{class}}{number\ of\ class\ * sample_{class}}$
- ☂ CrossEntropyLoss: weight = class_weights
- ☂ Cost-Sensitive: Penalize more on high weight class

Weather	Number	Weight
Cloudy	45,662	0.858
Rainy	1,369	28.6175
Sunny	70,501	0.5557

Training Results



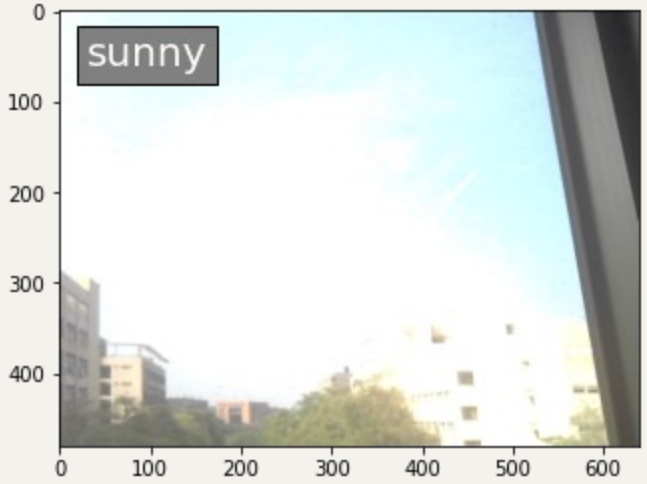


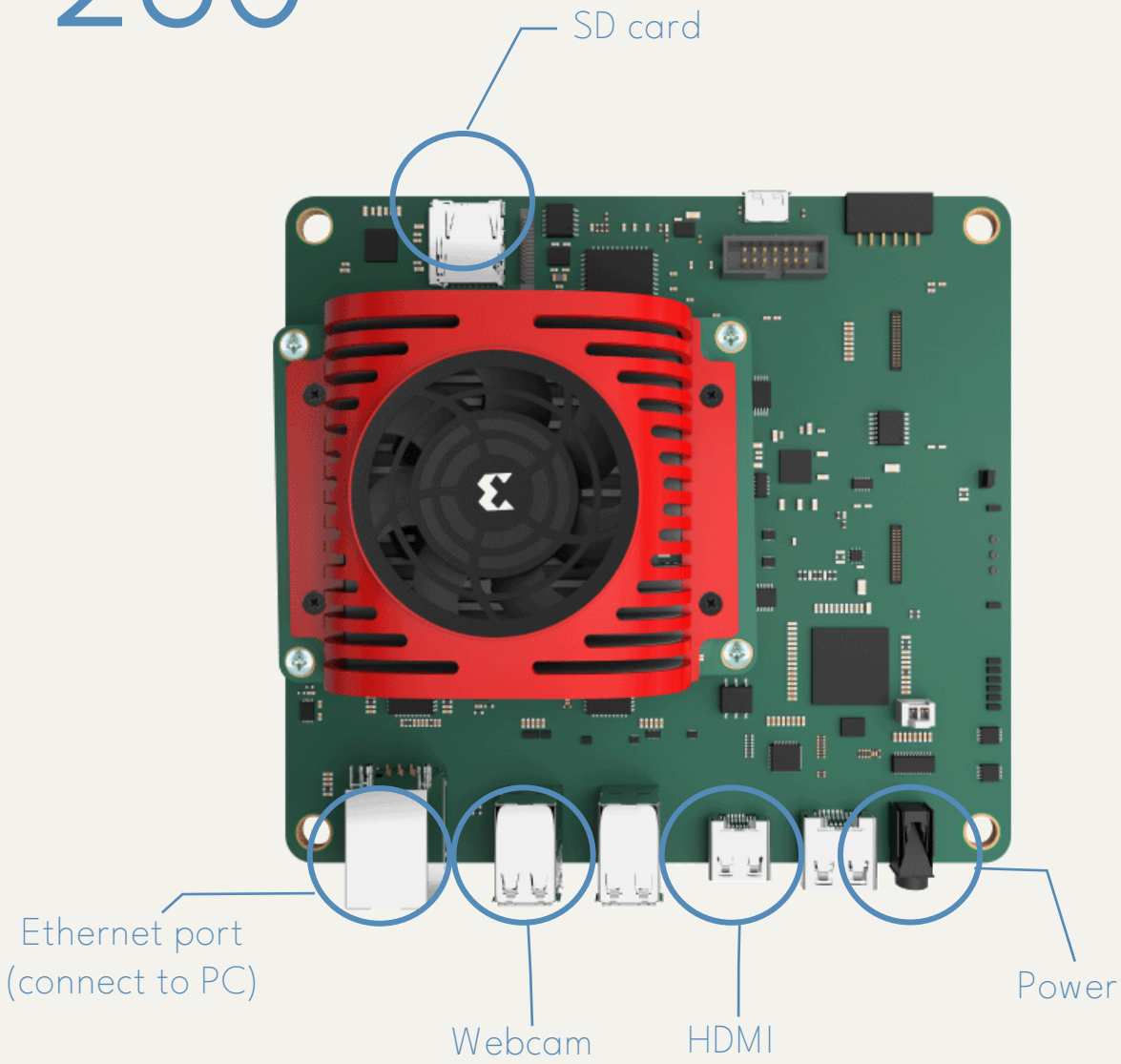
KV-260

- Connect to camera

  *Get real-time data

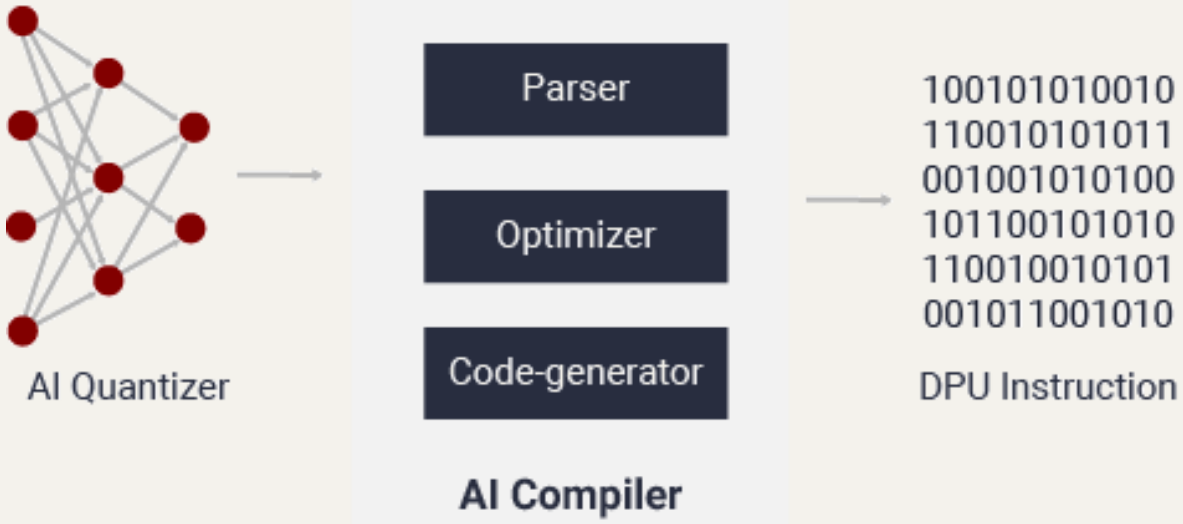
- Run the prediction



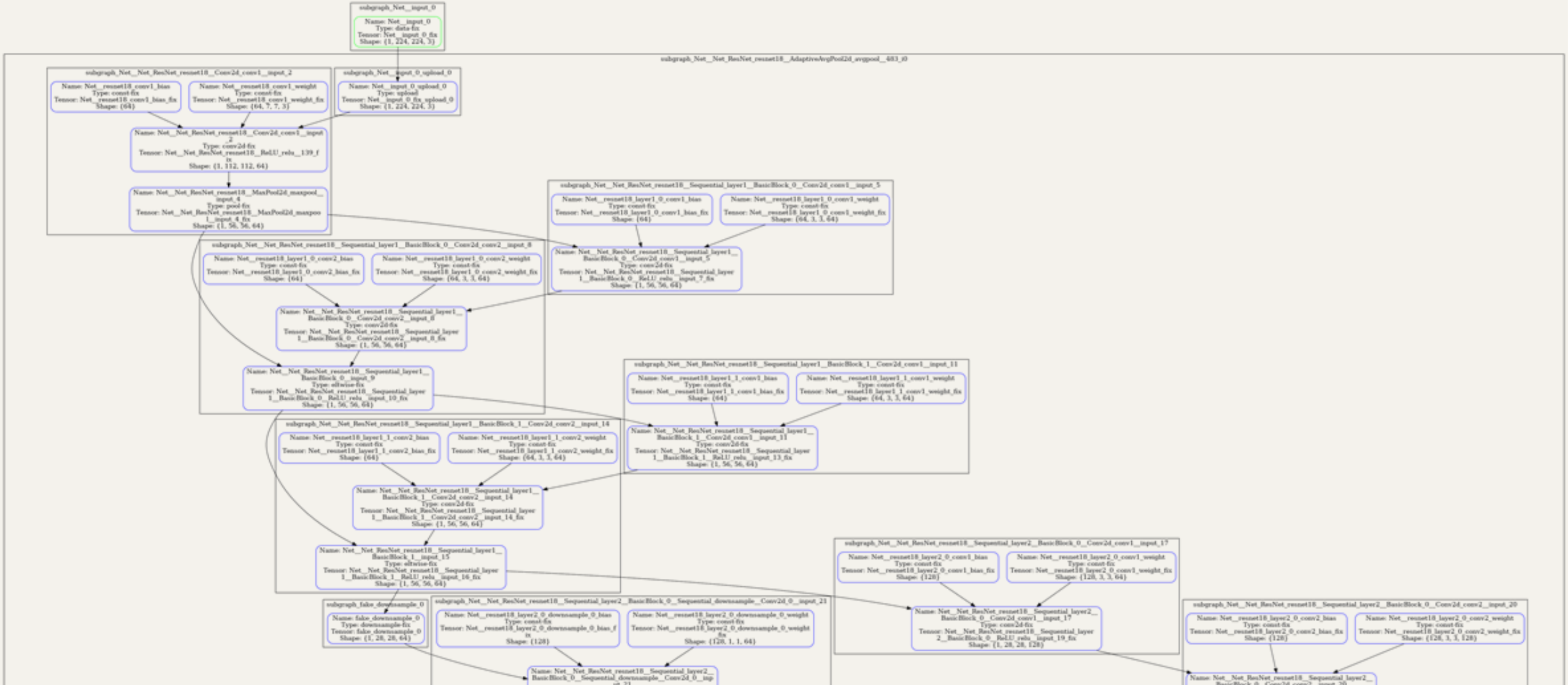


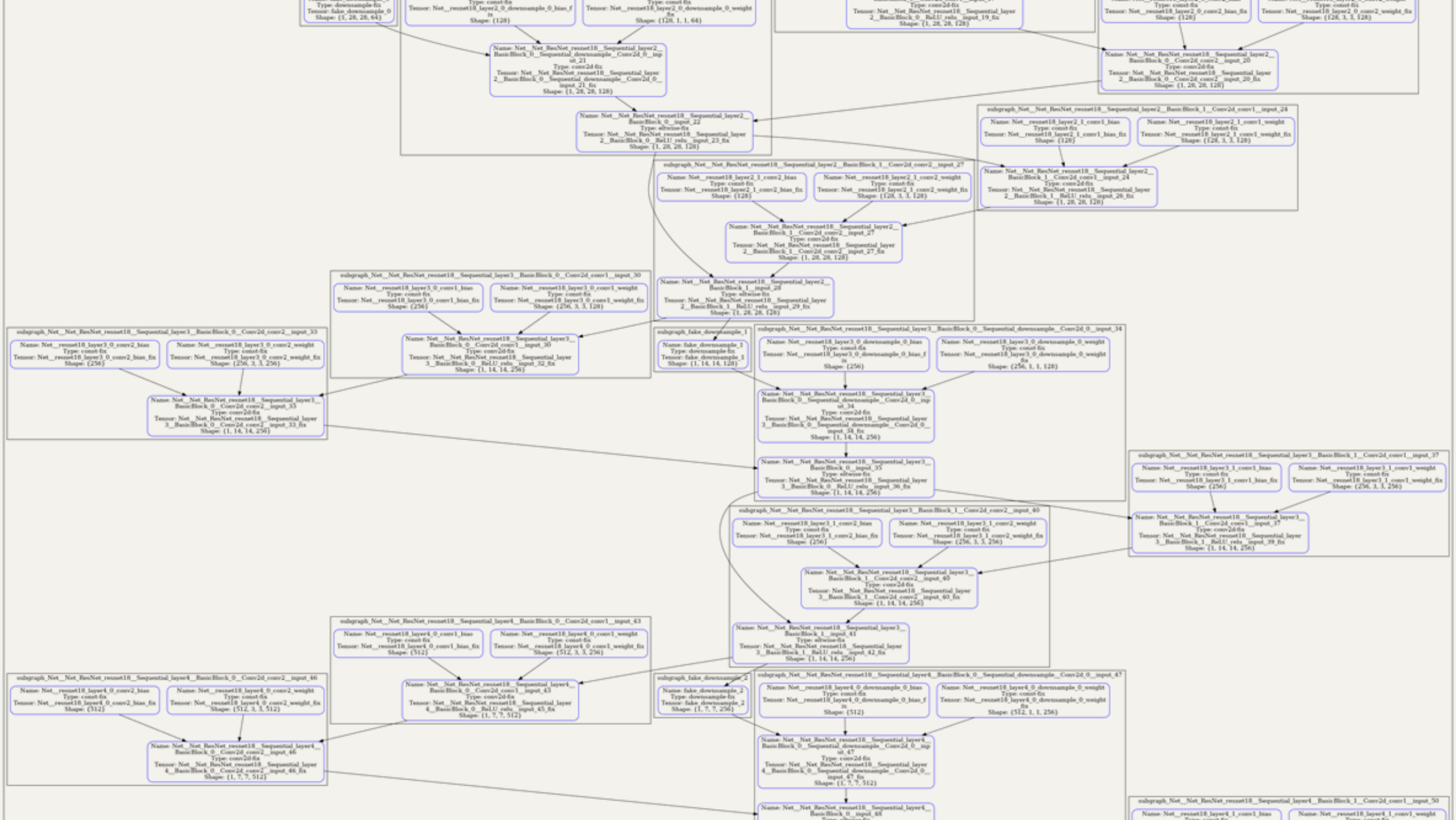
Vitis-AI

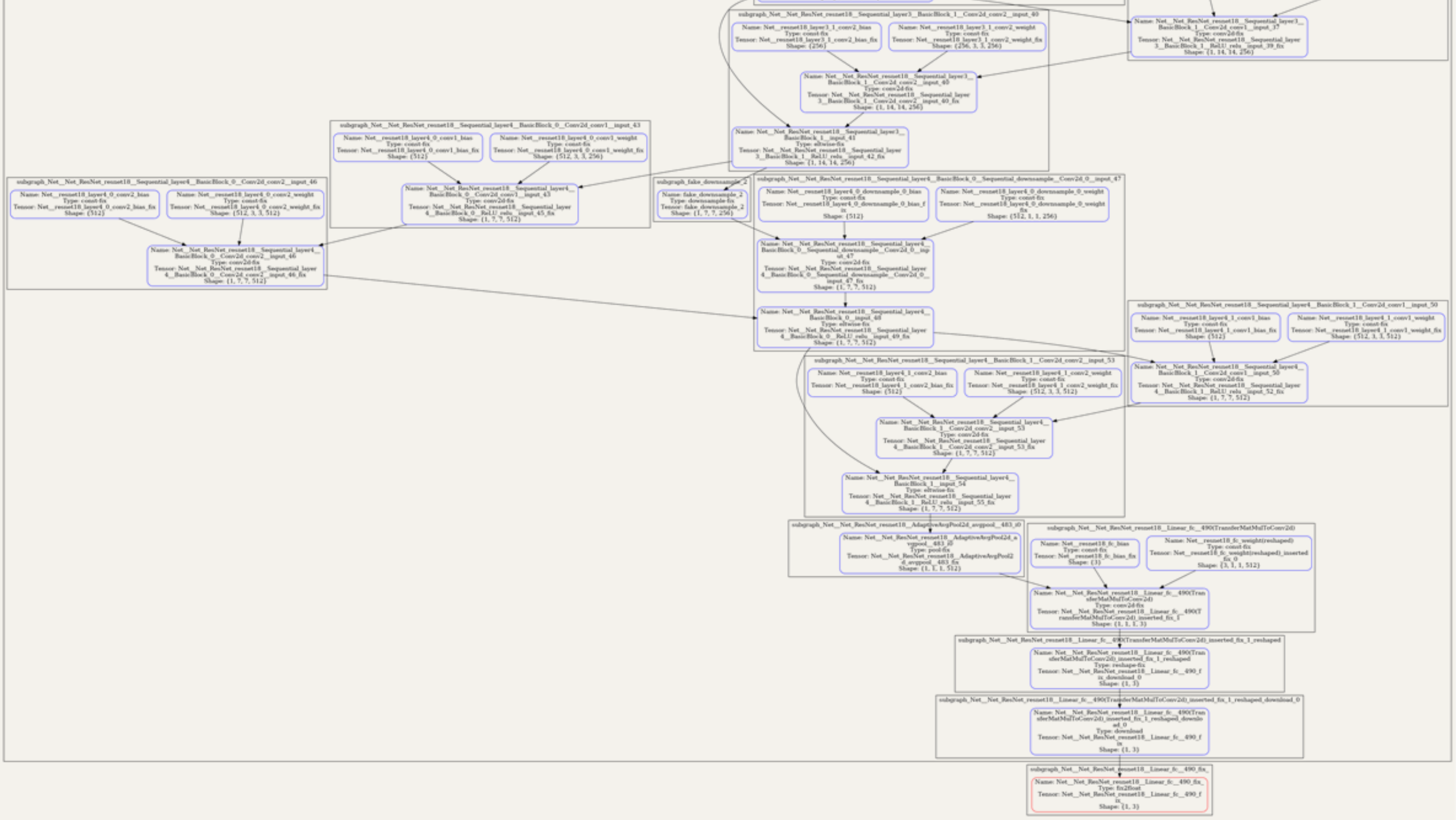
- Model Zoo
  - Pytorch: ResNet18
- AI Optimizer
  - Compress the model size
- AI Quantizer
  - Floating-point (32-bit) to fixed-point (8-bit)
  - Less memory bandwidth
- AI Compiler
  - Transforms the models into XIR-based computing graphs.
  - Breaks up the graph into several subgraphs on the basis of whether the operation can be executed on the DPU.
  - For the DPU subgraph, the compiler generates the high-efficient instruction stream.



Vitis-AI — Model Graph




subgraph_Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_0__Conv2d_conv2__input_20
Name: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_0__Conv2d_conv2__input_20
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_0__Conv2d_conv2__input_20_fix
Shape: {1, 28, 28, 128}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv1__input_24
Name: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv1__input_24
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv1__input_26_fix
Shape: {1, 28, 28, 128}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv2__input_27
Name: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv2__input_27
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer2__BasicBlock_1__Conv2d_conv2__input_27_fix
Shape: {1, 28, 28, 128}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Conv2d_conv1__input_30
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Conv2d_conv1__input_30
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__ReLU_relu__input_32_fix
Shape: {1, 14, 14, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Conv2d_conv2__input_33
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Conv2d_conv2__input_33
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Conv2d_conv2__input_33_fix
Shape: {1, 14, 14, 256}
subgraph_fake_downsample_1
Name: fake_downsample_1
Type: downsample-fix
Tensor: fake_downsample_1
Shape: {1, 14, 14, 128}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Sequential_downsample__Conv2d_0__input_34
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Sequential_downsample__Conv2d_0__input_34
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__Sequential_downsample__Conv2d_0__input_34_fix
Shape: {1, 14, 14, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__input_35
Type: eltwise-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_0__ReLU_relu__input_36_fix
Shape: {1, 14, 14, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv1__input_37
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv1__input_37
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__ReLU_relu__input_39_fix
Shape: {1, 14, 14, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40_fix
Shape: {1, 14, 14, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__input_41
Type: eltwise-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__ReLU_relu__input_42_fix
Shape: {1, 14, 14, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv1__input_43
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv1__input_43
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__ReLU_relu__input_45_fix
Shape: {1, 7, 7, 512}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46_fix
Shape: {1, 7, 7, 512}
subgraph_fake_downsample_2
Name: fake_downsample_2
Type: downsample-fix
Tensor: fake_downsample_2
Shape: {1, 7, 7, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47_fix
Shape: {1, 7, 7, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__input_48
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv1__input_50


subgraph_Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40
Name: Net__resnet18_layer3_1_conv2_bias
Type: const-fix
Tensor: Net__resnet18_layer3_1_conv2_bias_fix
Shape: {256}
Name: Net__resnet18_layer3_1_conv2_weight
Type: const-fix
Tensor: Net__resnet18_layer3_1_conv2_weight_fix
Shape: {256, 3, 3, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv2__input_40_fix
Shape: {1, 14, 14, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__input_41
Type: eltwise-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__ReLU_relu__input_42_fix
Shape: {1, 14, 14, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__Conv2d_conv1__input_37
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer3__BasicBlock_1__ReLU_relu__input_39_fix
Shape: {1, 14, 14, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv1__input_43
Name: Net__resnet18_layer4_0_conv1_bias
Type: const-fix
Tensor: Net__resnet18_layer4_0_conv1_bias_fix
Shape: {512}
Name: Net__resnet18_layer4_0_conv1_weight
Type: const-fix
Tensor: Net__resnet18_layer4_0_conv1_weight_fix
Shape: {512, 3, 3, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv1__input_43
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__ReLU_relu__input_45_fix
Shape: {1, 7, 7, 512}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46
Name: Net__resnet18_layer4_0_conv2_bias
Type: const-fix
Tensor: Net__resnet18_layer4_0_conv2_bias_fix
Shape: {512}
Name: Net__resnet18_layer4_0_conv2_weight
Type: const-fix
Tensor: Net__resnet18_layer4_0_conv2_weight_fix
Shape: {512, 3, 3, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Conv2d_conv2__input_46_fix
Shape: {1, 7, 7, 512}
subgraph_fake_downsample_2
Name: fake_downsample_2
Type: downsample-fix
Tensor: fake_downsample_2
Shape: {1, 7, 7, 256}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47
Name: Net__resnet18_layer4_0_downsample_0_bias
Type: const-fix
Tensor: Net__resnet18_layer4_0_downsample_0_bias_fix
Shape: {512}
Name: Net__resnet18_layer4_0_downsample_0_weight
Type: const-fix
Tensor: Net__resnet18_layer4_0_downsample_0_weight_fix
Shape: {512, 1, 1, 256}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__Sequential_downsample__Conv2d_0__input_47_fix
Shape: {1, 7, 7, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__input_48
Type: eltwise-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_0__ReLU_relu__input_49_fix
Shape: {1, 7, 7, 512}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv1__input_50
Name: Net__resnet18_layer4_1_conv1_bias
Type: const-fix
Tensor: Net__resnet18_layer4_1_conv1_bias_fix
Shape: {512}
Name: Net__resnet18_layer4_1_conv1_weight
Type: const-fix
Tensor: Net__resnet18_layer4_1_conv1_weight_fix
Shape: {512, 3, 3, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv1__input_50
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__ReLU_relu__input_52_fix
Shape: {1, 7, 7, 512}
subgraph_Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv2__input_53
Name: Net__resnet18_layer4_1_conv2_bias
Type: const-fix
Tensor: Net__resnet18_layer4_1_conv2_bias_fix
Shape: {512}
Name: Net__resnet18_layer4_1_conv2_weight
Type: const-fix
Tensor: Net__resnet18_layer4_1_conv2_weight_fix
Shape: {512, 3, 3, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv2__input_53
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__Conv2d_conv2__input_53_fix
Shape: {1, 7, 7, 512}
Name: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__input_54
Type: eltwise-fix
Tensor: Net__Net_ResNet_resnet18__Sequential_layer4__BasicBlock_1__ReLU_relu__input_55_fix
Shape: {1, 7, 7, 512}
subgraph_Net__Net_ResNet_resnet18__AdaptiveAvgPool2d_avgpool__483_i0
Name: Net__Net_ResNet_resnet18__AdaptiveAvgPool2d_avgpool__483_i0
Type: pool-fix
Tensor: Net__Net_ResNet_resnet18__AdaptiveAvgPool2d_avgpool__483_fix
Shape: {1, 1, 1, 512}
subgraph_Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)
Name: Net__resnet18_fc_bias
Type: const-fix
Tensor: Net__resnet18_fc_bias_fix
Shape: {3}
Name: Net__resnet18_fc_weight(reshaped)
Type: const-fix
Tensor: Net__resnet18_fc_weight(reshaped)_inserted_fix_0
Shape: {3, 1, 1, 512}
Name: Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)
Type: conv2d-fix
Tensor: Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)_inserted_fix_1
Shape: {1, 1, 1, 3}
subgraph_Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)_inserted_fix_1_reshaped
Name: Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)_inserted_fix_1_reshaped
Type: reshape-fix
Tensor: Net__Net_ResNet_resnet18__Linear_fc__490_fix_download_0
Shape: {1, 3}
subgraph_Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)_inserted_fix_1_reshaped_download_0
Name: Net__Net_ResNet_resnet18__Linear_fc__490(TransferMatMulToConv2d)_inserted_fix_1_reshaped_download_0
Type: download
Tensor: Net__Net_ResNet_resnet18__Linear_fc__490_fix
Shape: {1, 3}
subgraph_Net__Net_ResNet_resnet18__Linear_fc__490_fix_
Name: Net__Net_ResNet_resnet18__Linear_fc__490_fix_
Type: fix2float
Tensor: Net__Net_ResNet_resnet18__Linear_fc__490_fix
Shape: {1, 3}

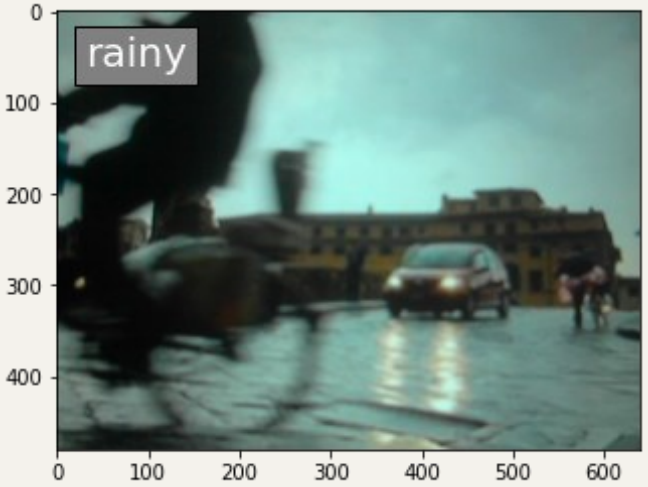
Deploy and Run the Model

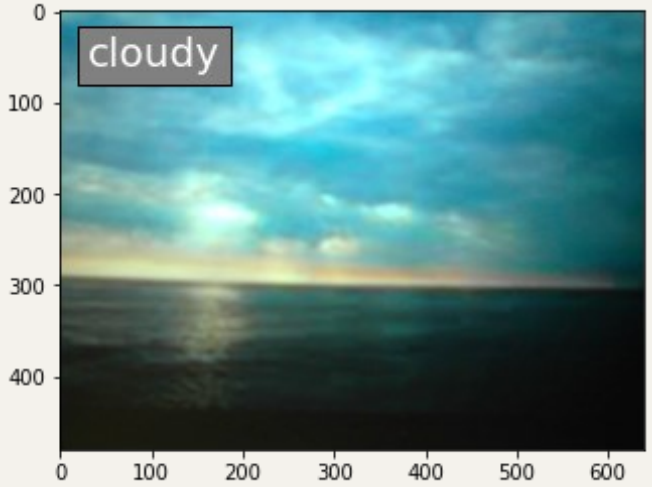
```
overlay.load_model("group13_kv260.xmodel")

dpu = overlay.runner ------------------------------------- Return: An instance of DPU runner

inputTensors = dpu.get_input_tensors() ------------- Return: A list of DPU runner inputs
outputTensors = dpu.get_output_tensors() ---------------Return: A vector of raw pointer
                                                                    to the output tensor

job_id = dpu.execute_async(input_data, output_data)-------------- Executes the runner
dpu.wait(job_id) ------------------------------------------------------ Waits for the end of DPU
                                                                    processing
```

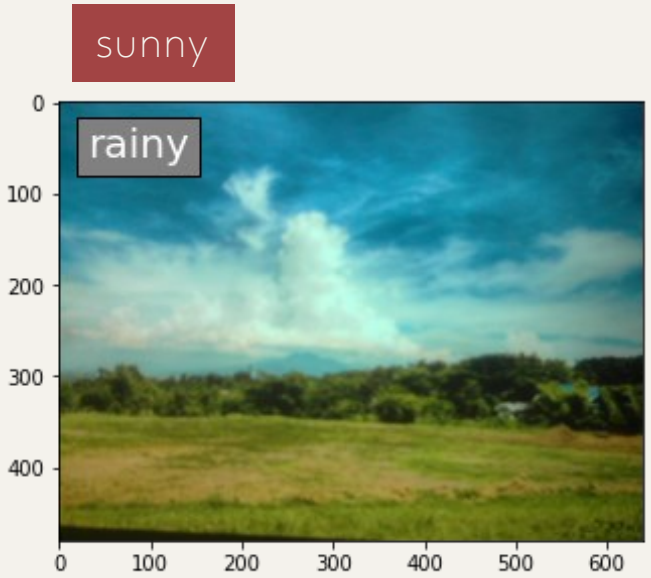
Demo

Image2Weather dataset

*Data is shown on the screen of the notebook.



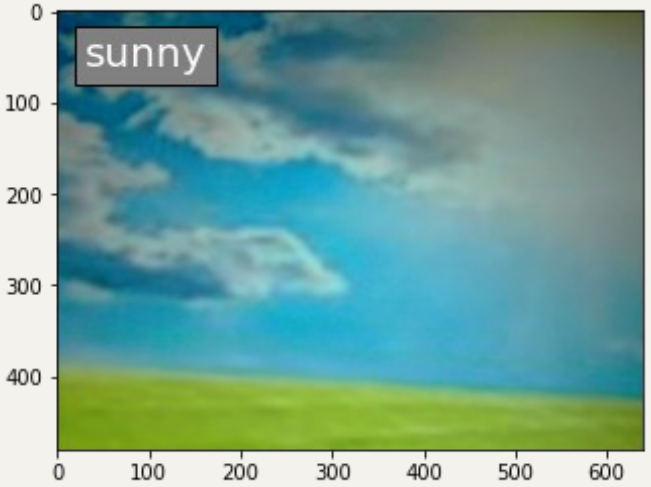


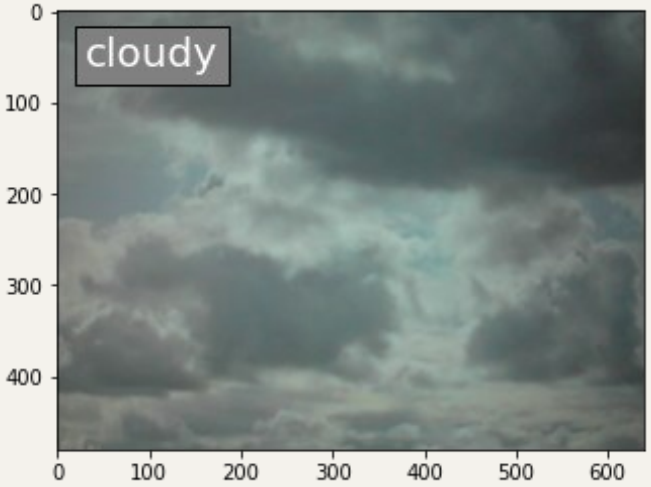


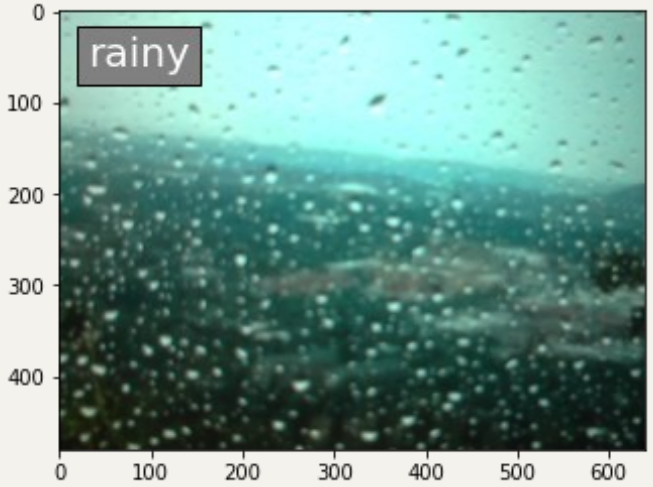
Demo

☃ Multi-class weather dataset (MWD)

*Data is shown on the screen of the notebook.



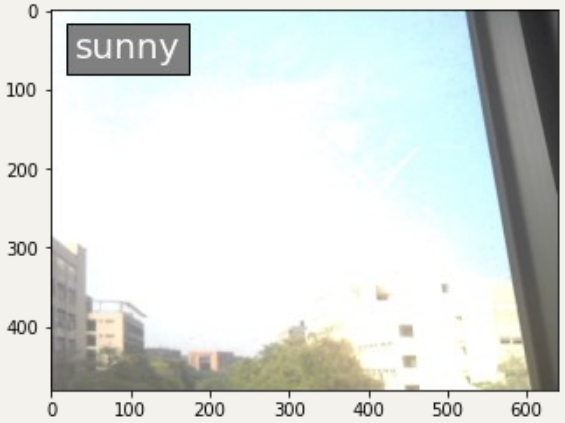


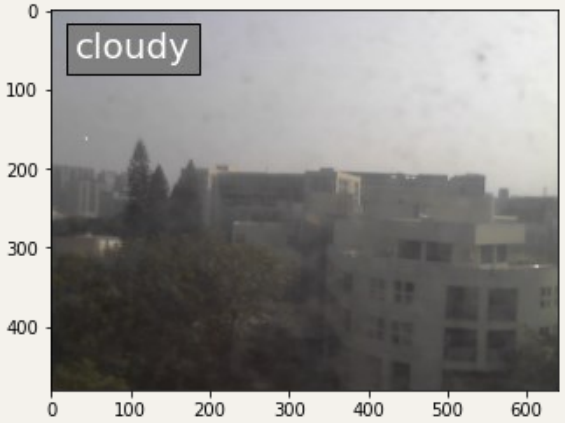


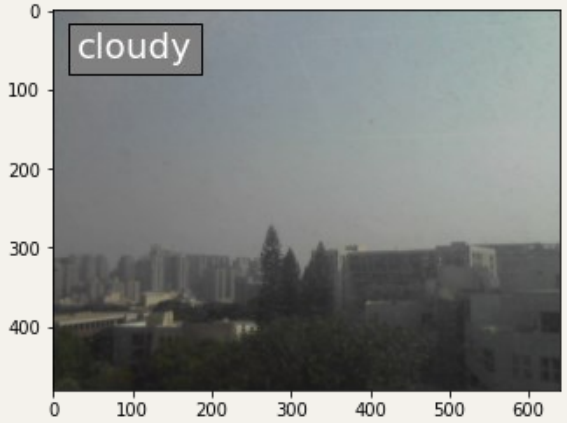
Demo

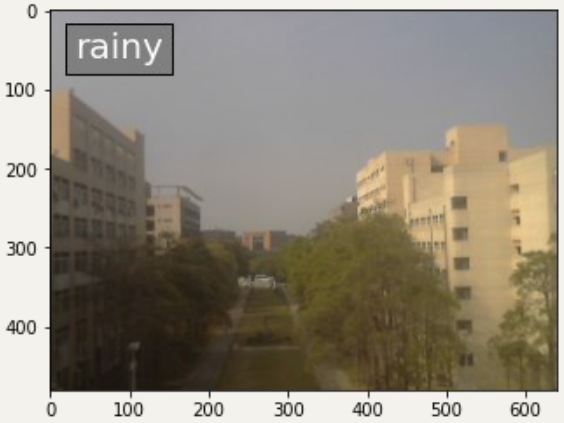
☃ Real scenarios (Library)

*Indoor, Dirty glasses ☹



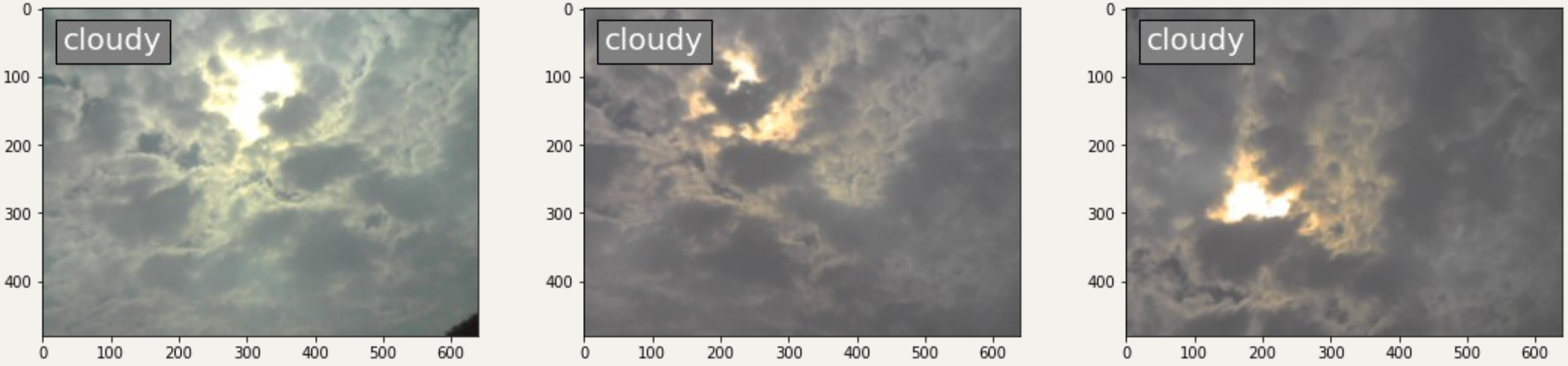






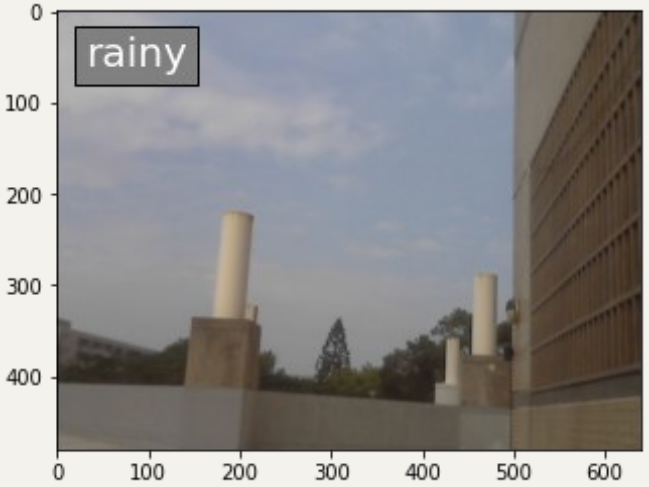
Demo

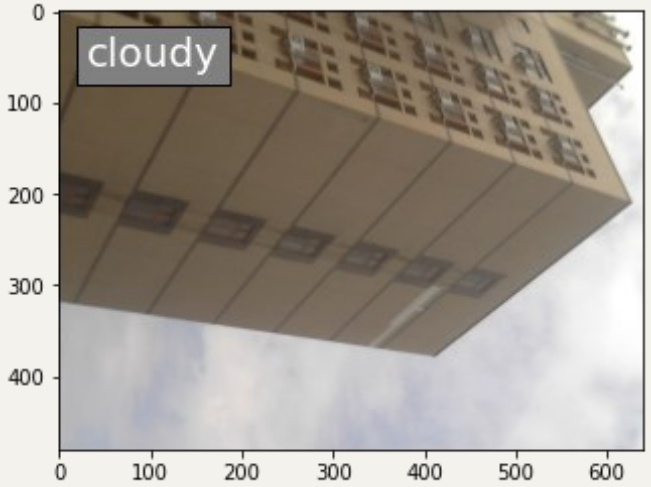
☃ Real scenarios (Girls Dorm 2)



Demo

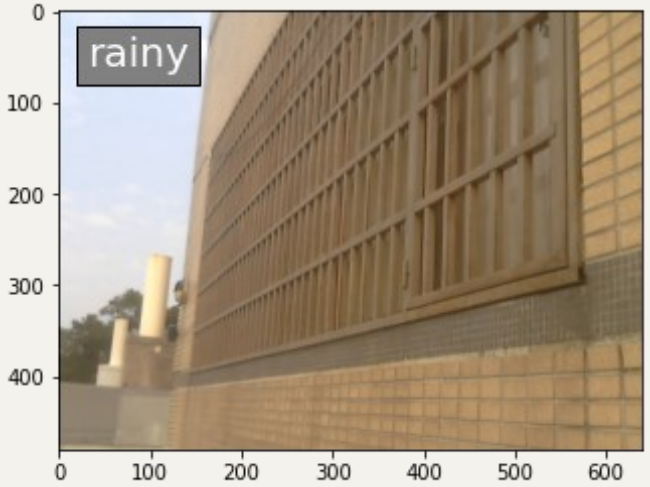
☃ Real scenarios (Girls Dorm 2)

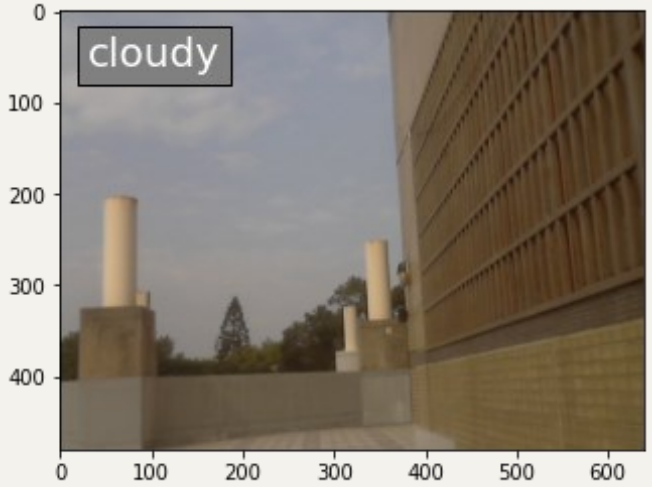


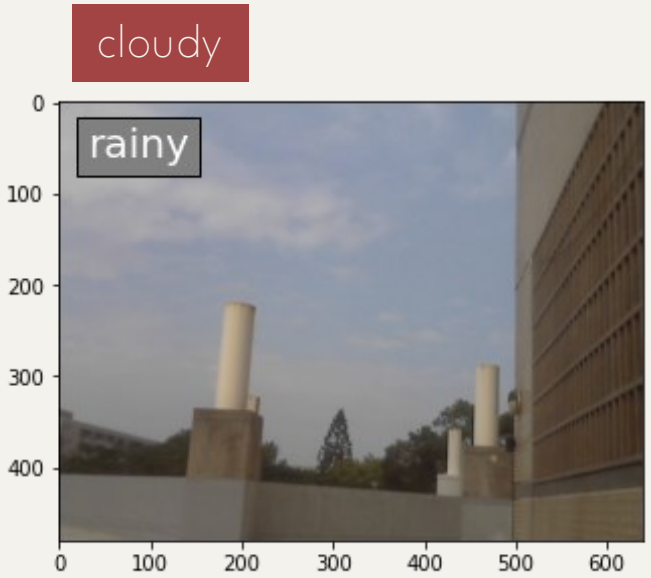


Demo

Real scenarios (Girls Dorm 2)



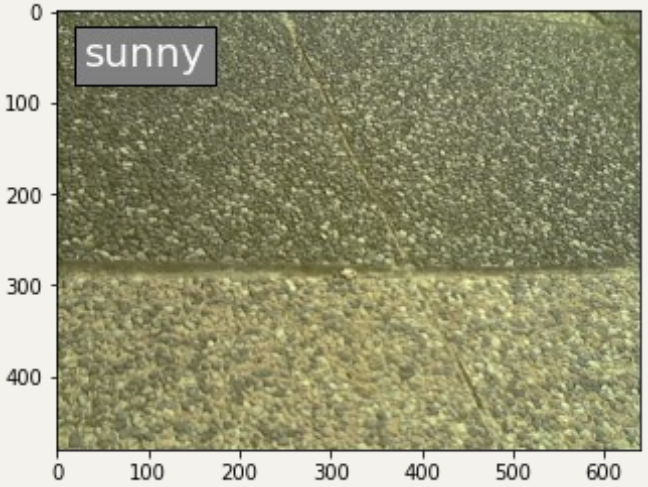


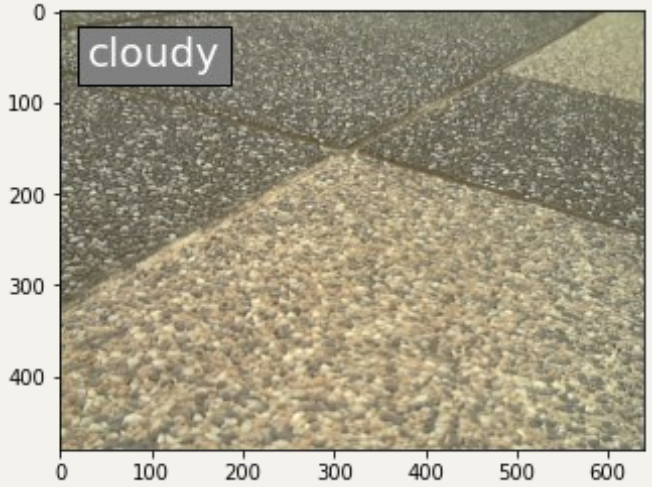


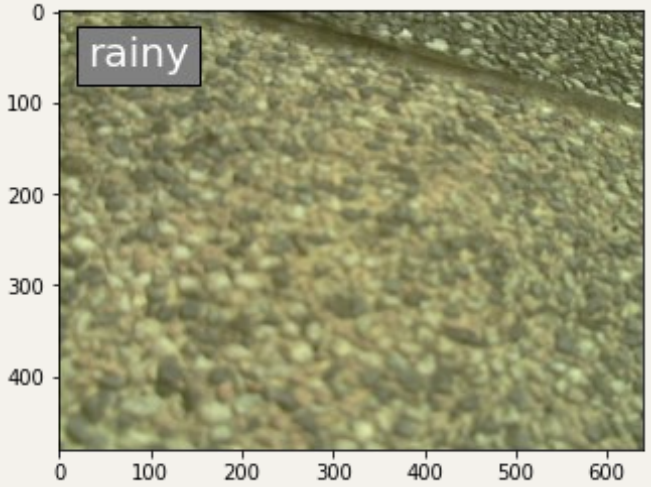
Demo

Real scenarios (Girls Dorm 2)

*Floor, unable to classify







Obstacles

404
Not Found

Obstacles

- While Testing our works
  - Unable to bring the system outside the building
    - → Use notebook to display the pictures
- Testing in the library
  - The windows is too dirty
  - The weather is the same most of the time
- The light might affect the result

Future Development

- Try more data resampling methods
  - Over/Under Sampling
  - Hybrid Methods
  - Add more classes
- Train a bigger model
  - Model Pruning/Distillation
  - Dual Mutual Learning
- Data Augmentation/Domain adaption
- Use continuous images for regconition

Q & A

Thank You