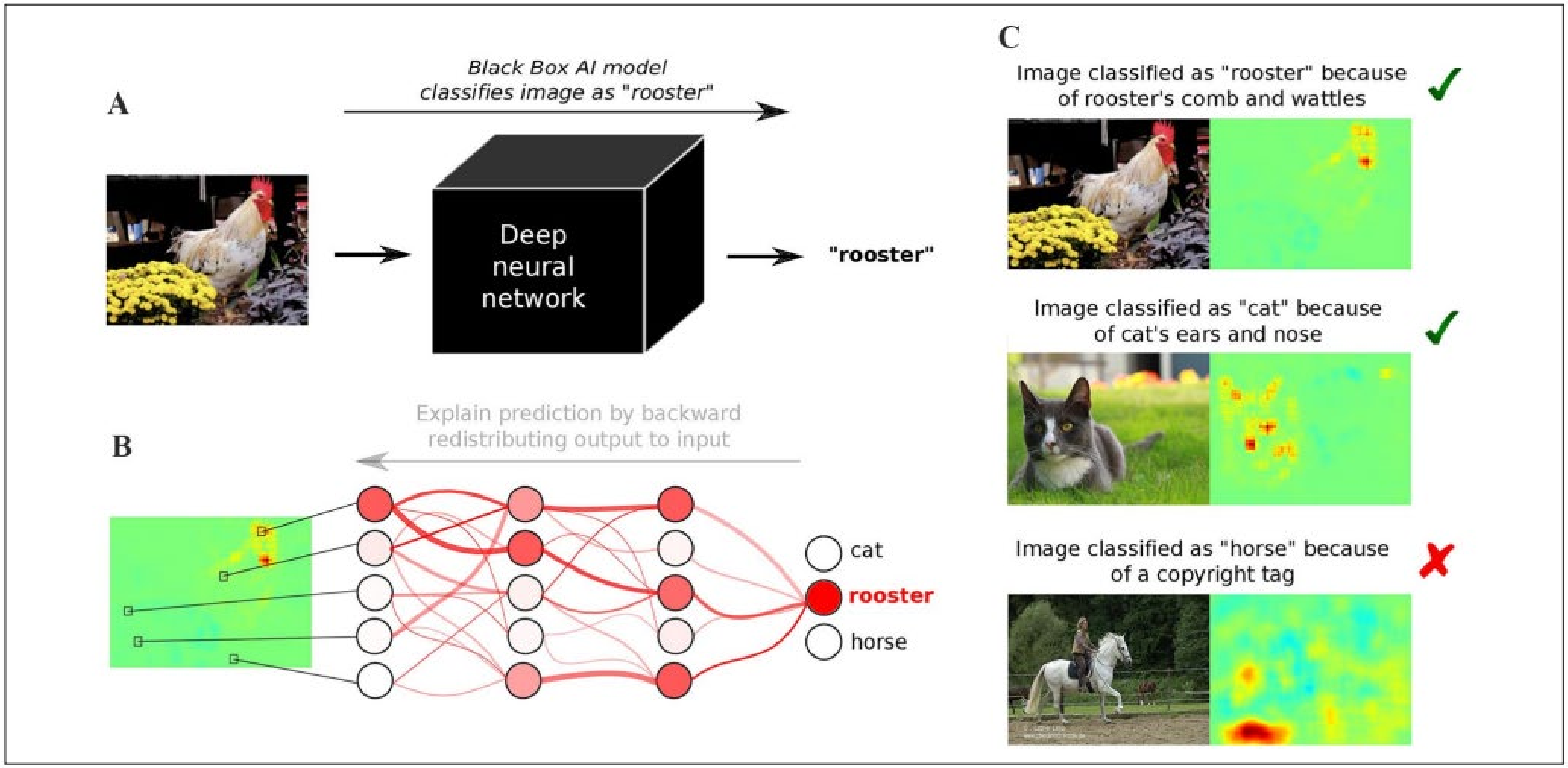
XAI

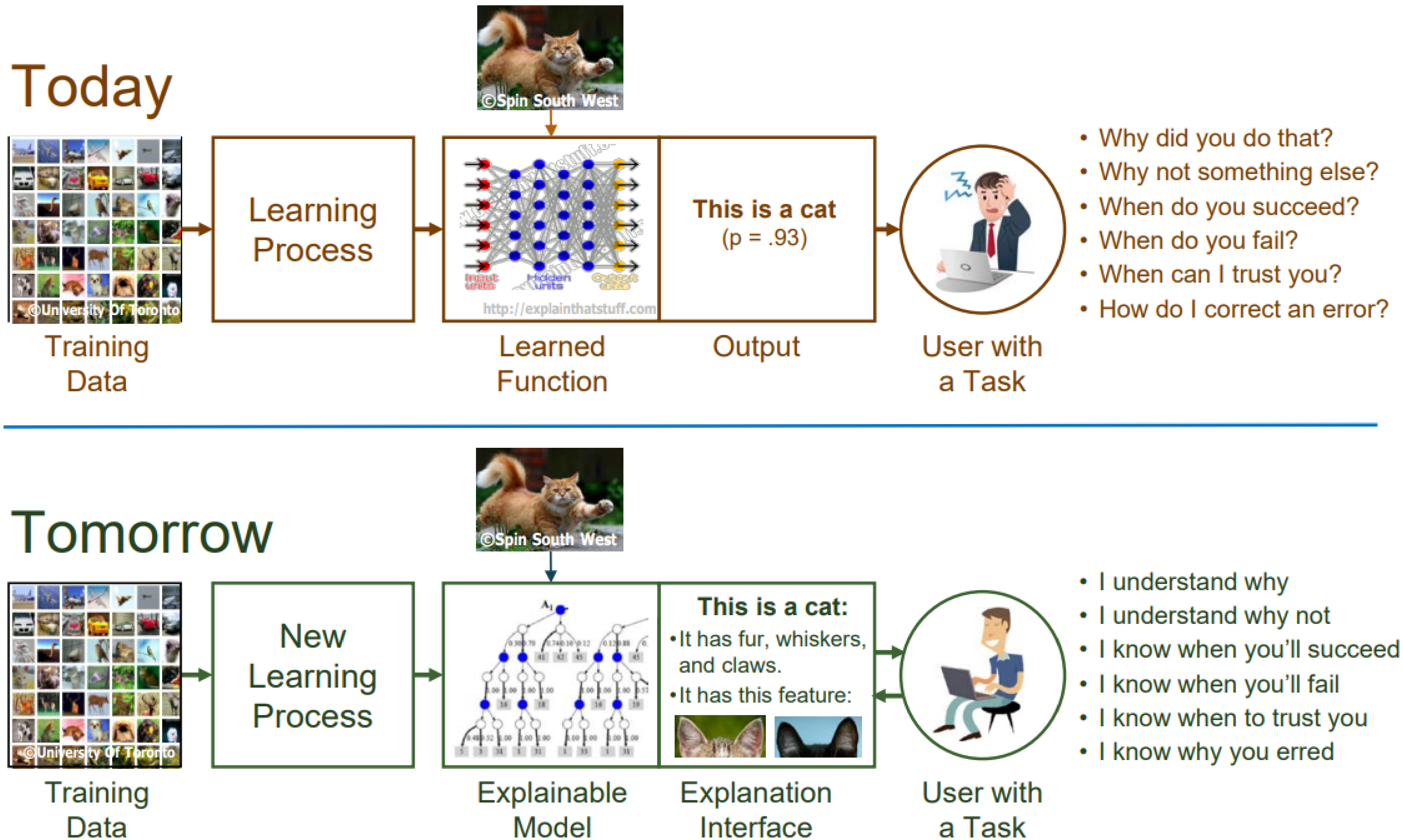
TA. Bogyeong Suh

Black-box model



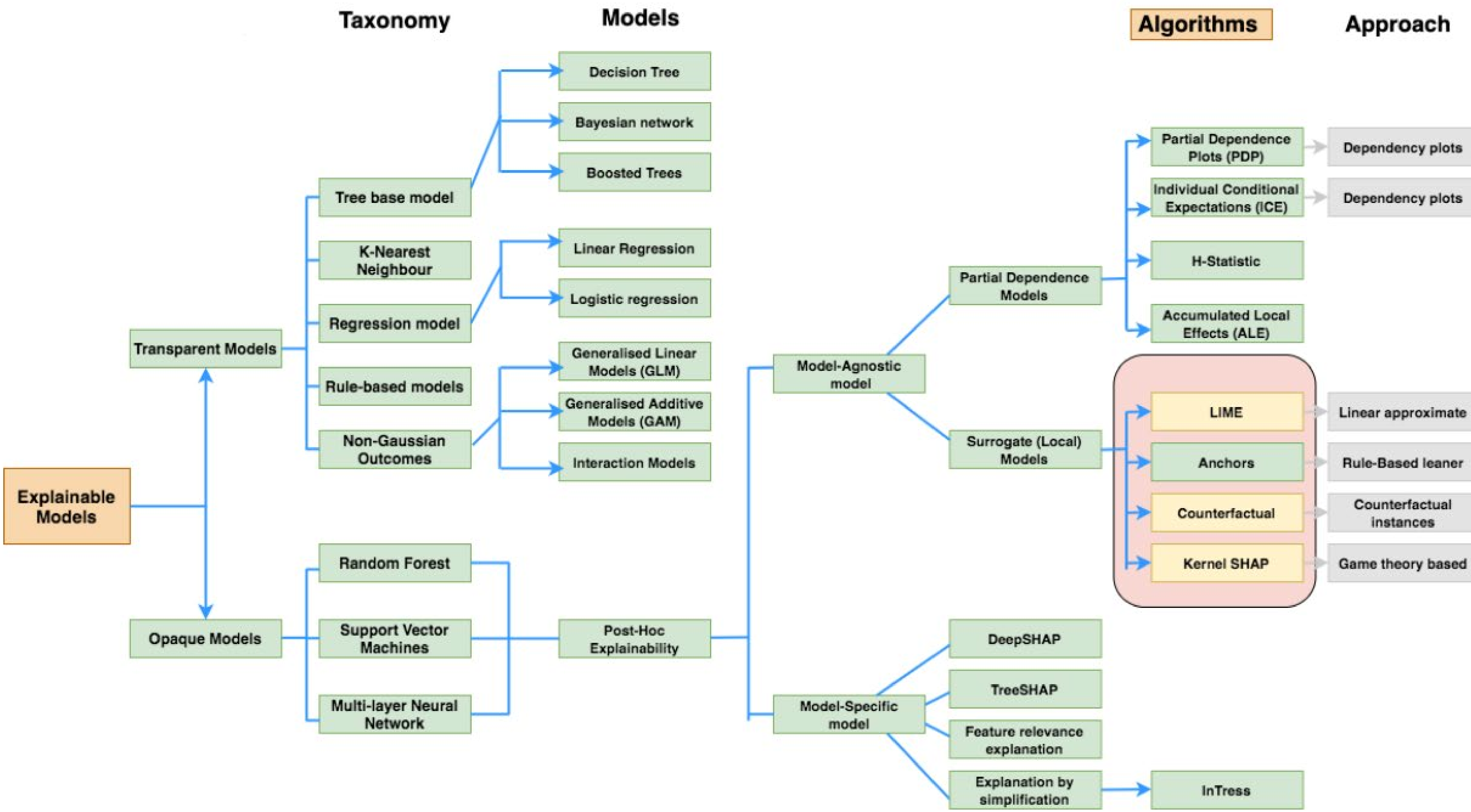
- Internal workings and decision-making processes are not easily understood or transparent in black-box models
- This makes it difficult to understand the reasoning behind the model’s decisions and to identify any potential biases or errors in its predictions

Black-box model



- **Explainability** and **interpretability** are important features that reflect the reliability and practicality of AI models

XAI(eXplainable Artificial Intelligence)



- AI in which humans can understand the decisions or predictions made by the AI
- XAI provides clear and interpretable reasons of the AI models' decision-making processes

Types of XAI

Intrinsic vs Post-hoc

- Intrinsic: model with an interpretable structure. It may have low accuracy due to less complexity in the structure.
- Post-hoc: applied after a model has been trained and deployed.

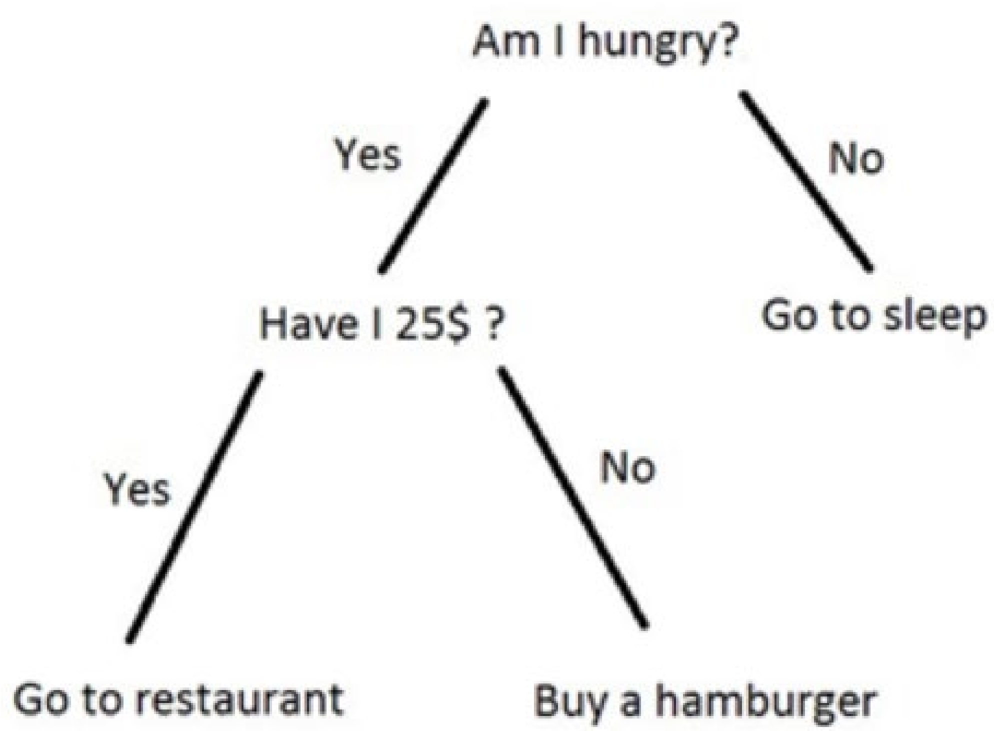
Global vs Local

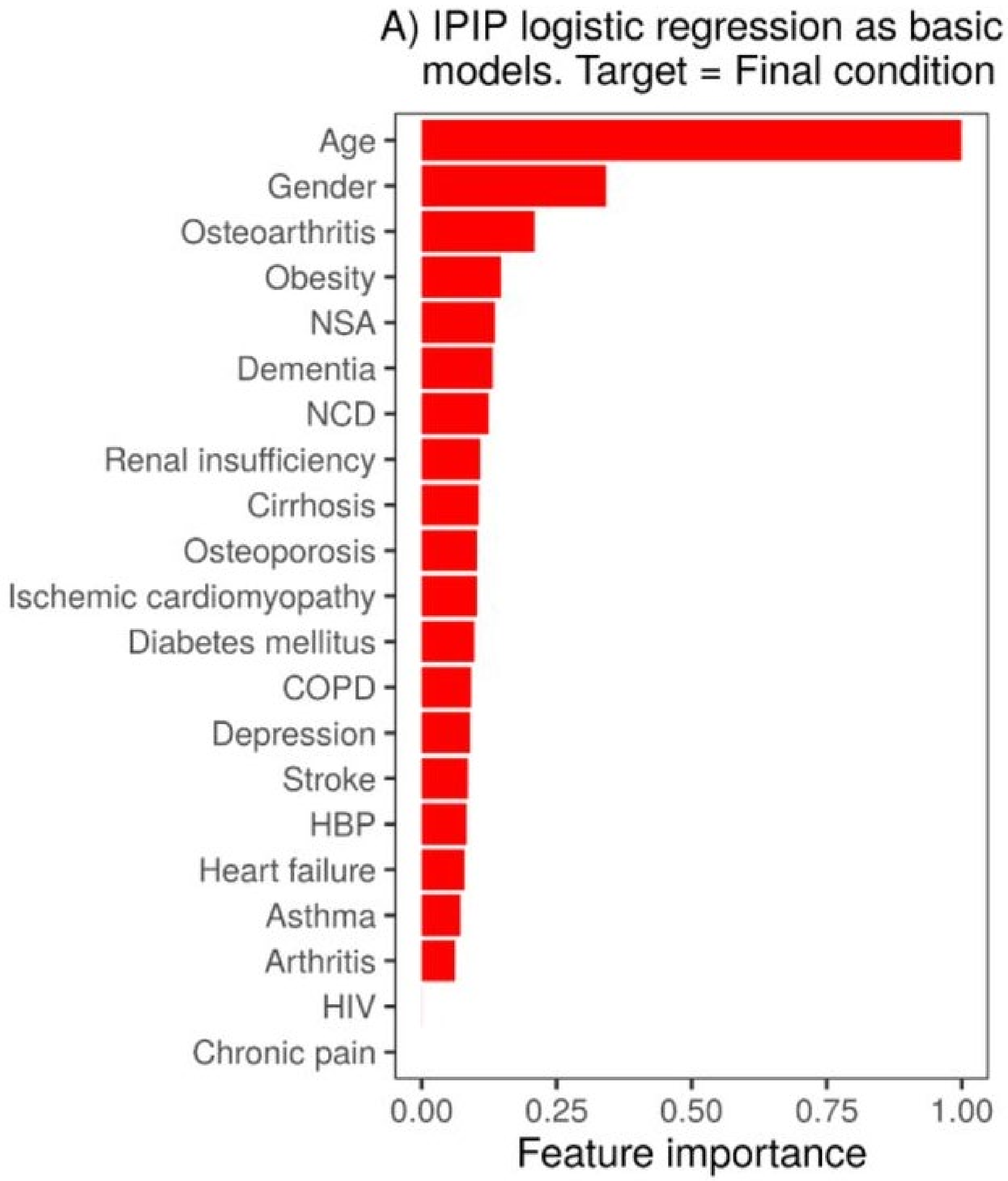
- Global: explains every predictions that a model makes.
- Local: explains certain decision mechanism or one prediction.

Model-specific vs Model-agnostic

- Model-specific: can be applied to certain types of models.
- Model-agnostic: can be applied to any types of models.

Model-intrinsic XAI(Interpretable model)



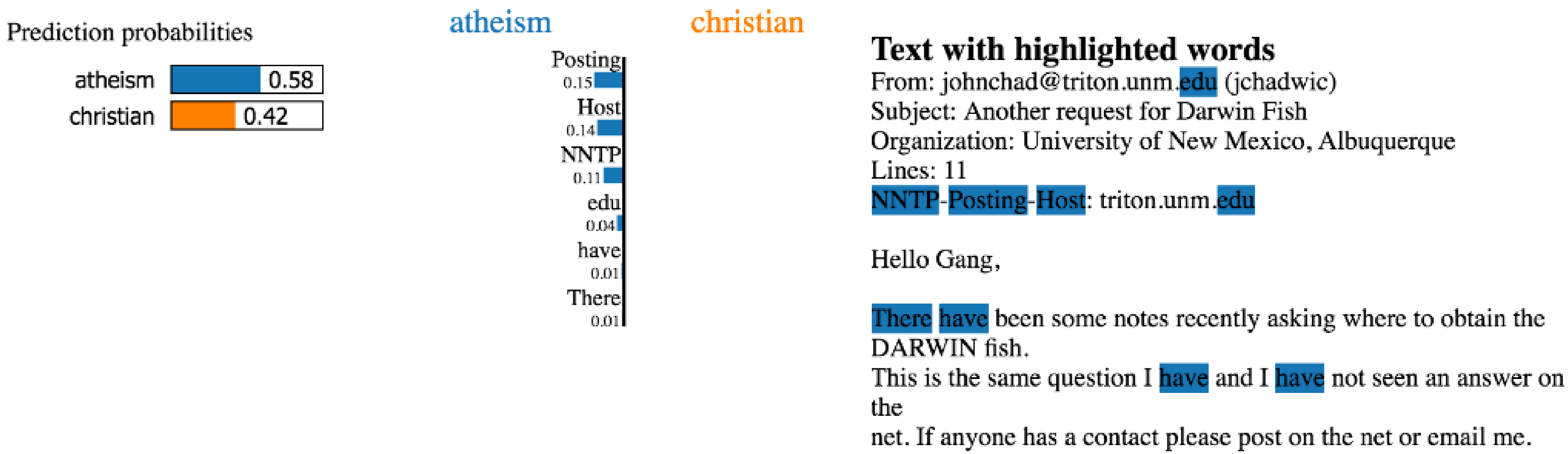


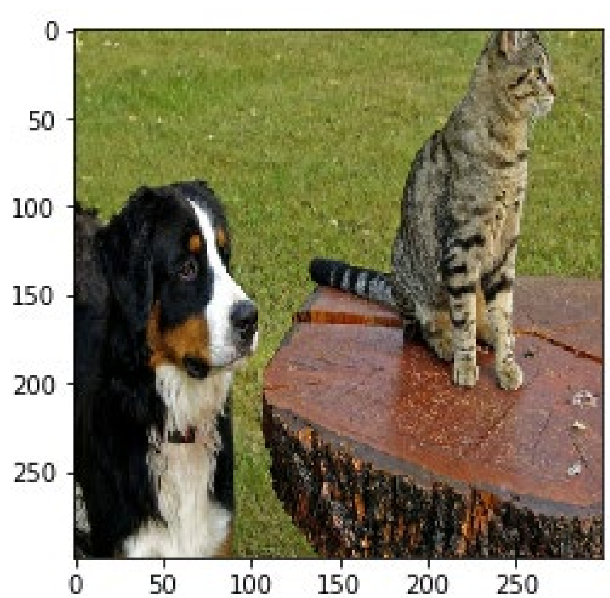
The model itself is explainable

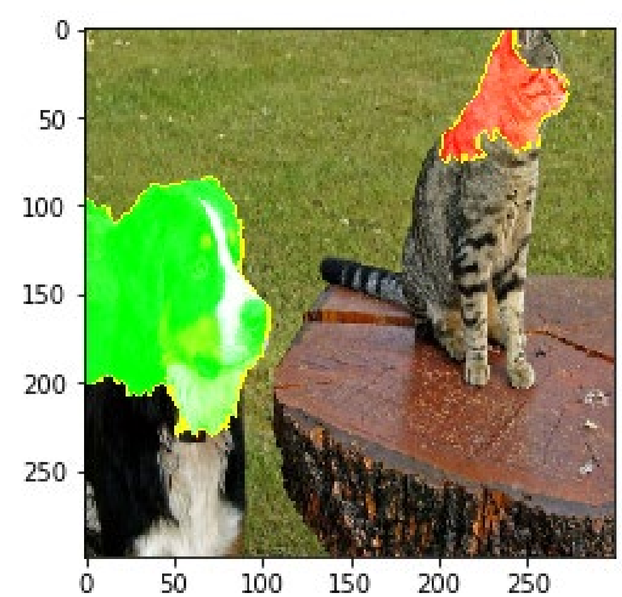
1) Decision Tree: Process of deriving the prediction results and its interpretation are intuitive

2) Linear model: Coefficients of the model can be used to determine the importance of each features in the prediction (Logistic Regression, Support Vector Machines, LASSO regression, etc)

LIME(Local Interpretable Model-agnostic Explanation)







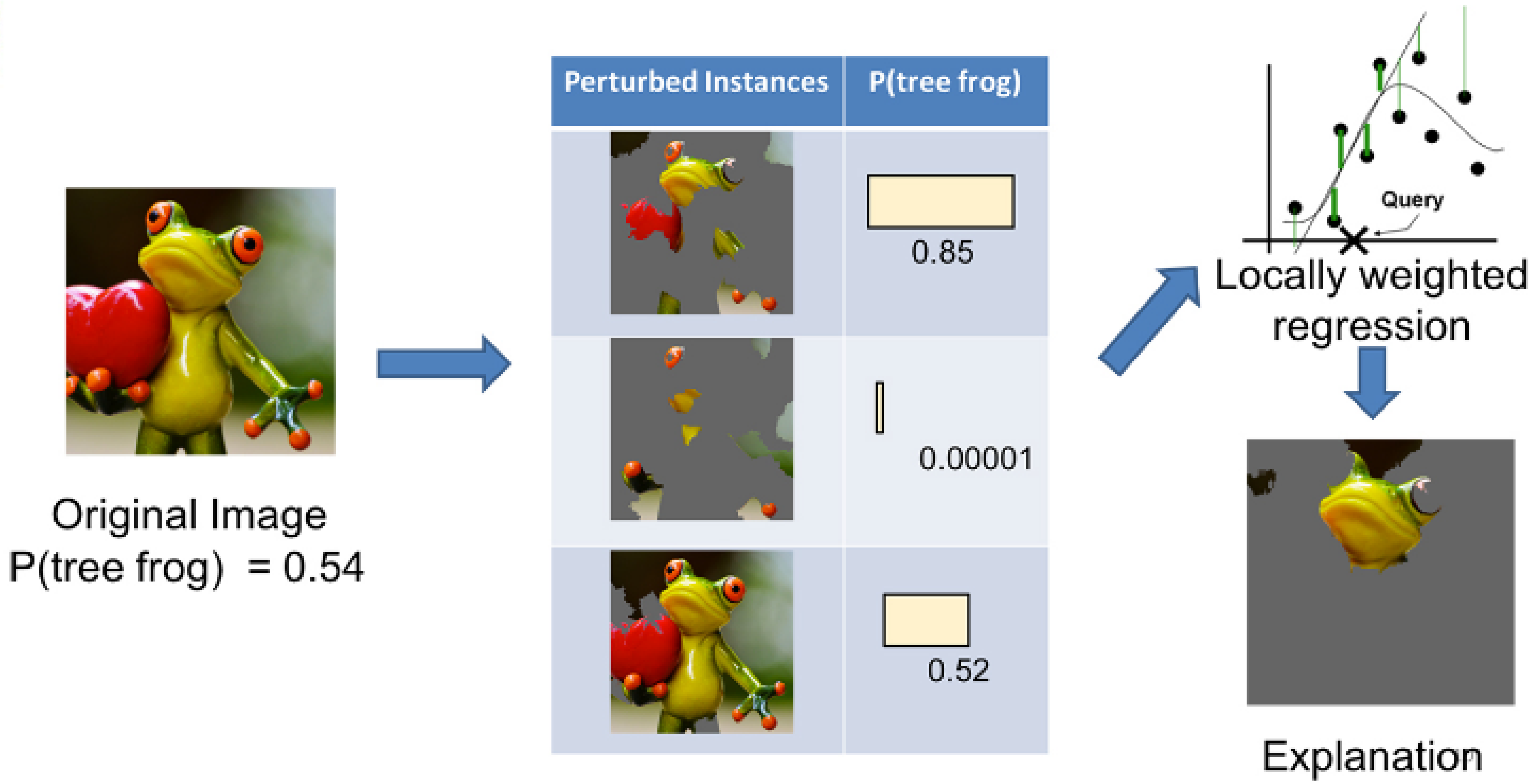
- Model-agnostic, and provide local interpretability by explaining individual predictions
- LIME provides explanation by permuting the observation(dataset) and watching the resulting predictions from the model
- Text data, tabular data, images in classification

LIME(Local Interpretable Model-agnostic Explanation)

e.g) image classification


Original Image


Interpretable Components



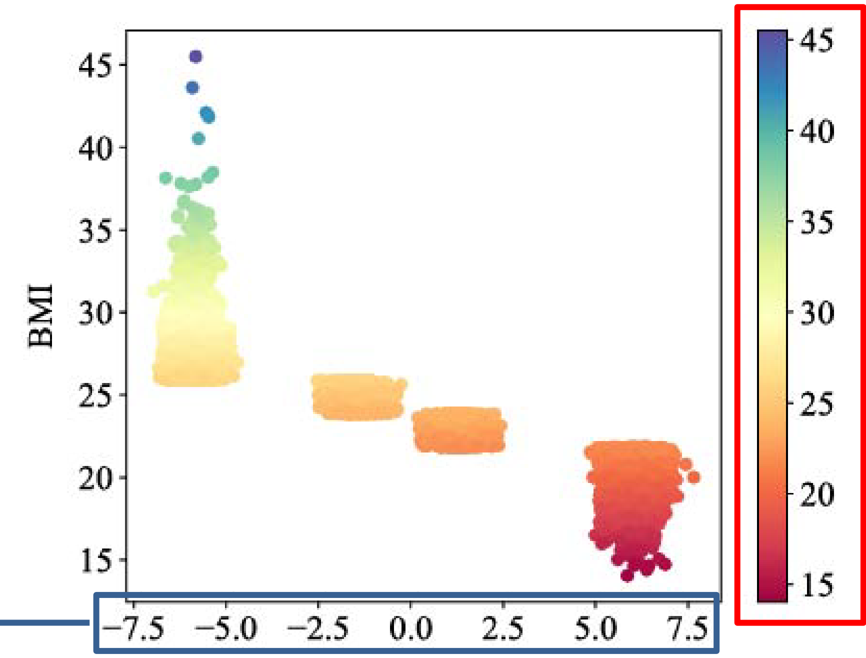
- Perturbation can be made by making superpixels as gray components
- If the prediction results change a lot, then perturbed pixels are the important components when making the prediction

LIME(Local Interpretable Model-agnostic Explanation)
e.g) feature contribution rank

Probability of being classified into osteoporosis

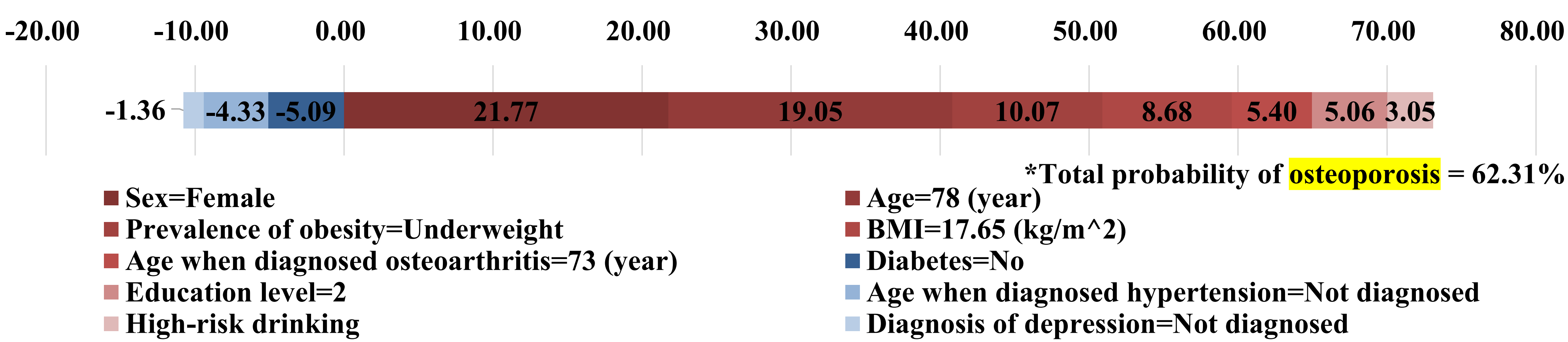
−7.5 −5.0 −2.5 0.0 2.5 5.0 7.5

Normal Osteoporosis



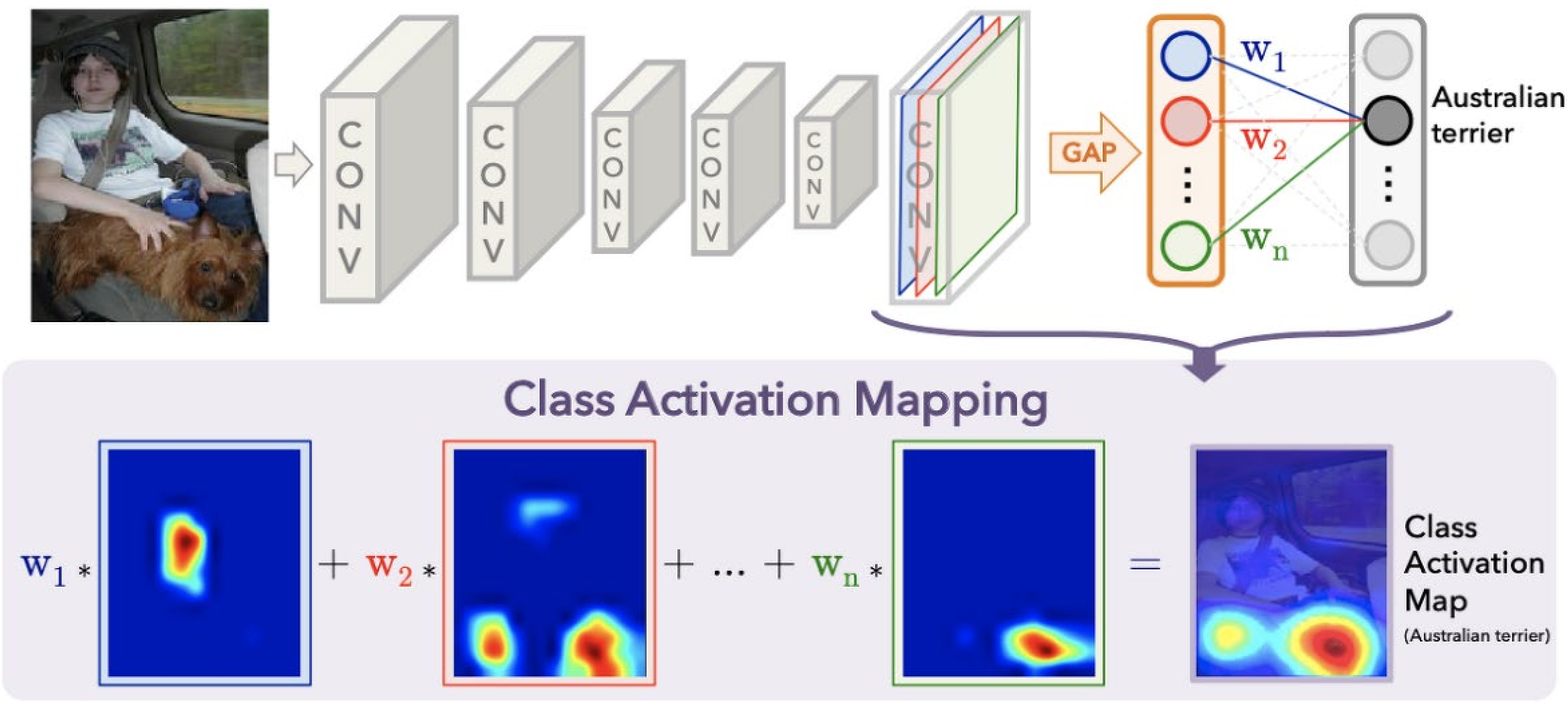
Individualized risk assessment for osteoporosis

Model prediction for osteoporosis (%)



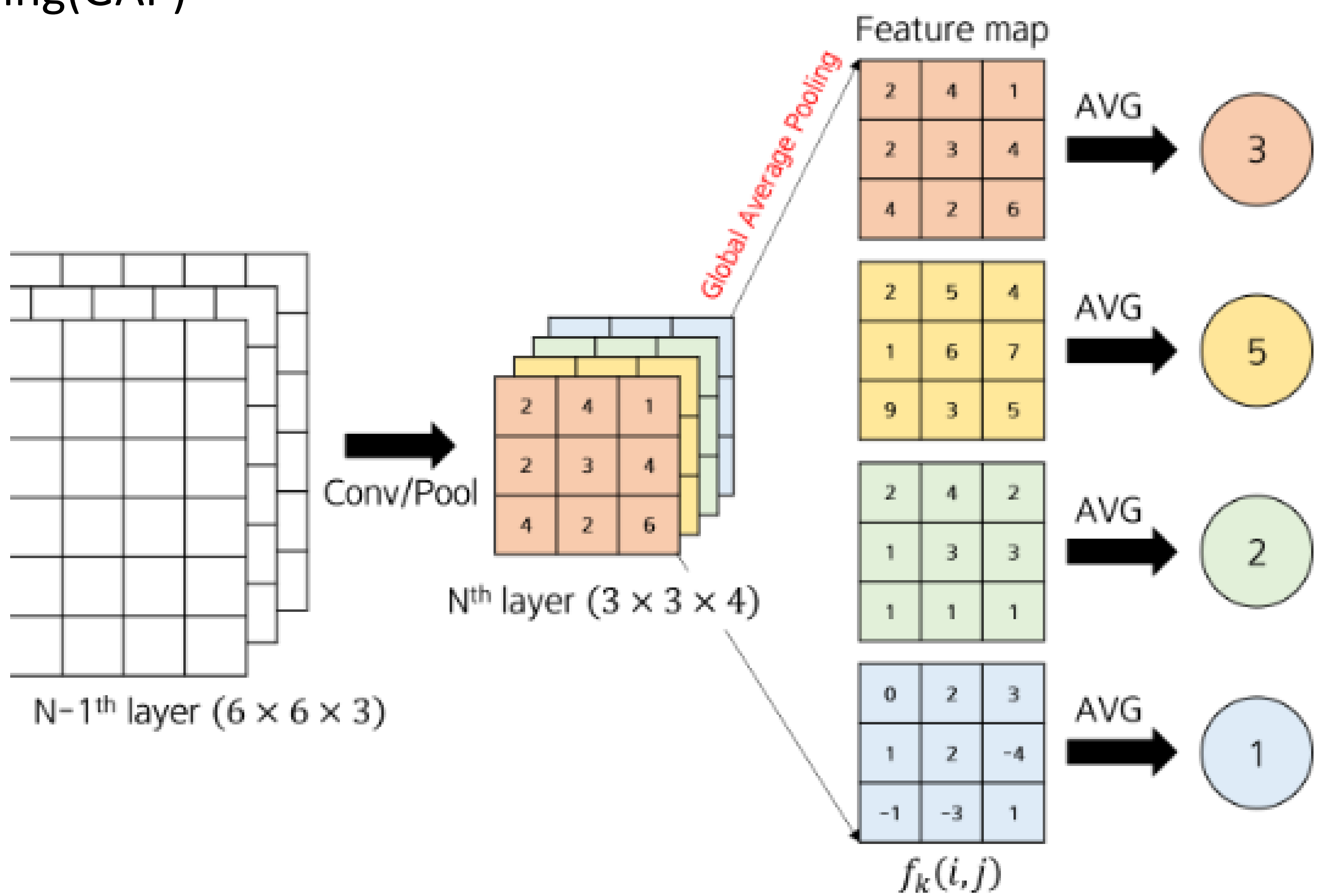
- Feature contribution rank can be made with the analysis of feature importance from LIME

CAM(Class Activation Map)



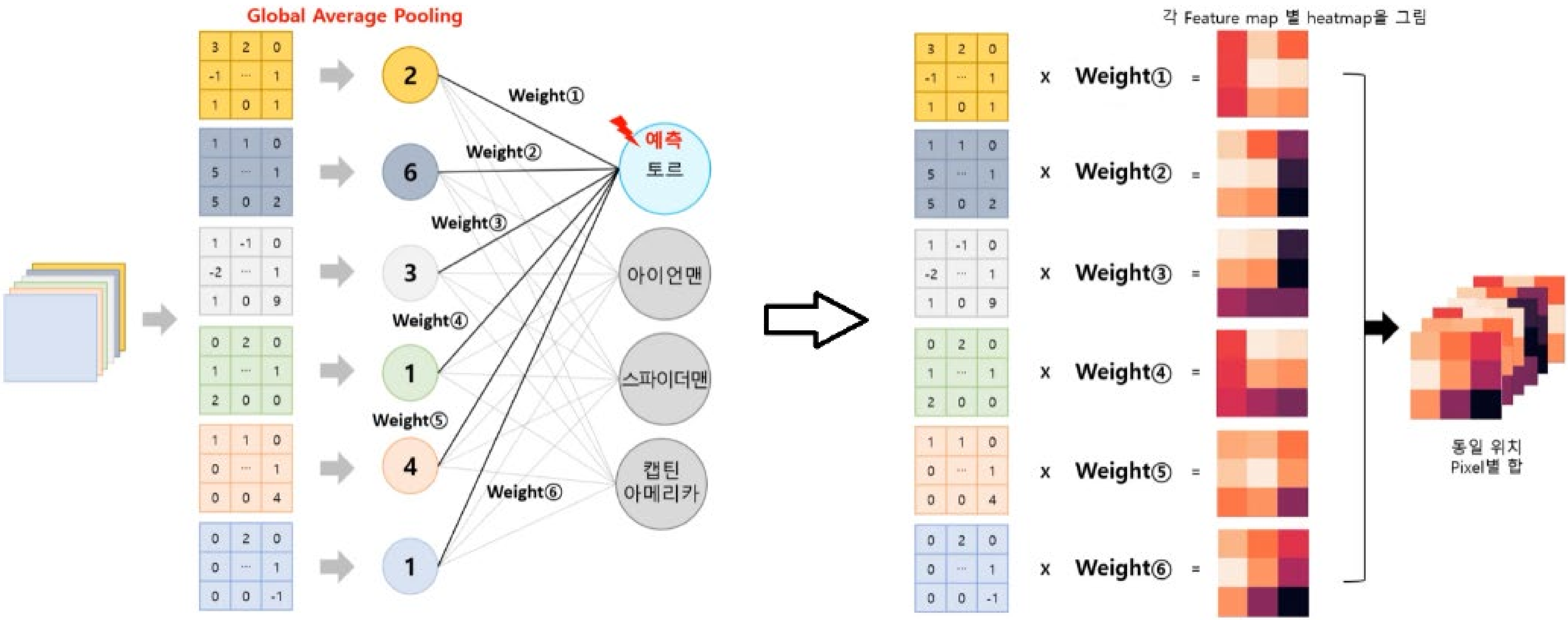
- CAM is a type of visualization in deep learning to understand the regions of an image that are important for a particular classification decision made by a Convolutional Neural Network(CNN)

CAM(Class Activation Map)
-Global Average Pooling(GAP)



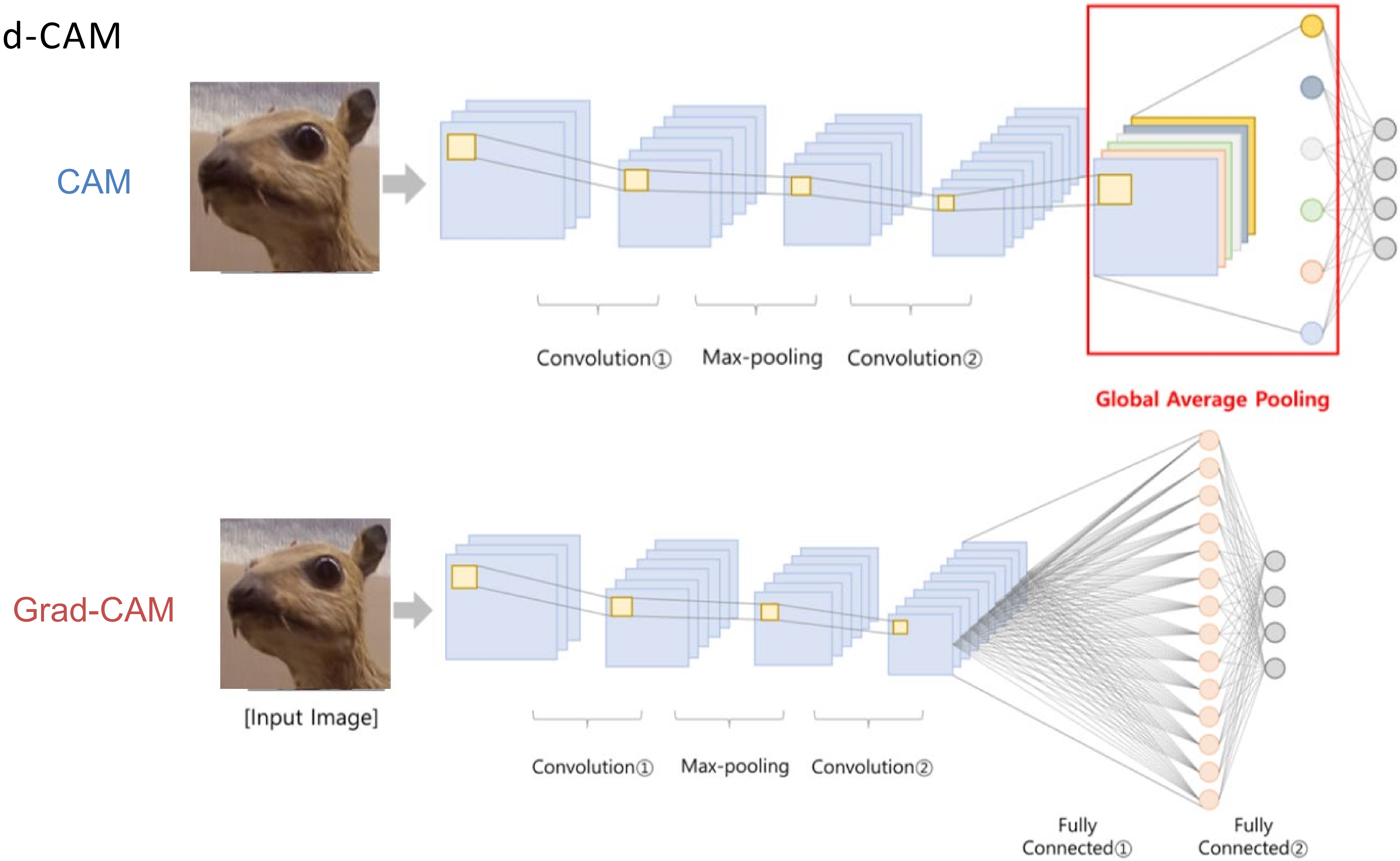
- Instead of flattening the data with FC(Fully Connected) layer in the last layer of CNN, using GAP(Global Average Pooling) can generate a heatmap that highlights the regions of the input image that are most relevant to the prediction
- With GAP, feature maps of each channel are converted into a vector with the length equivalent to the number of channels

CAM(Class Activation Map)



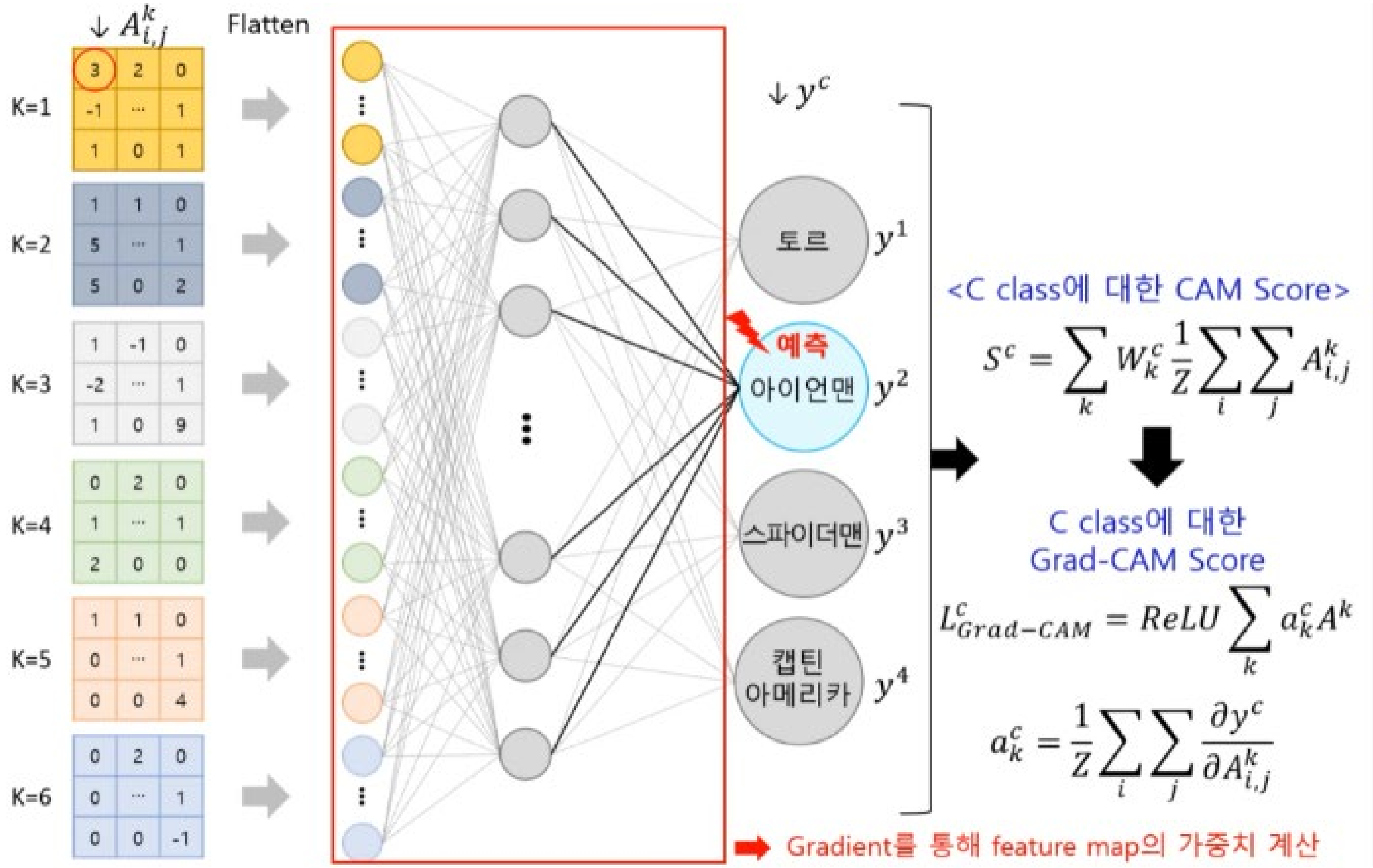
- The output of GAP layer is given as input data to the FC layer which is attached right after the GAP layer
- Heatmap can be made by multiplying each feature map with the weight and then adding to each pixels
- Training is needed to find the weight of FC layer

Grad-CAM



- CAM is only applicable with CNN performing GAP over convolutional maps immediately prior to prediction (heatmap can be printed out only from the feature maps of the last convolution layer)
- Grad-CAM has similar structure to the original CNN (two FC layers without GAP), there is no need to retrain the CNN model (weight can be derived with gradient information)

Grad-CAM



- The neuron importance weights derived from gradients are multiplied with feature maps
- ReLU function is applied to ensure that the final CAM only highlights the regions of the image that are positively contributing to the model's decision

CAM on Cat vs Dogs classification CNN model

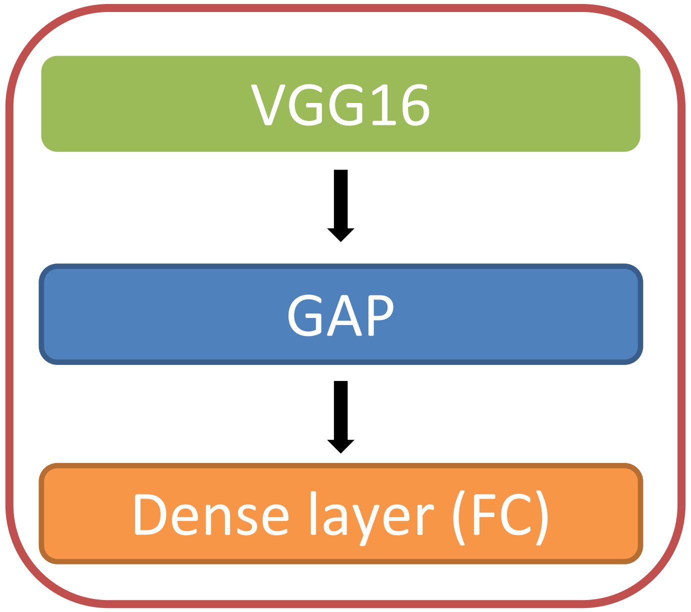
```python
base_model = models.vgg16(weights=models.VGG16_Weights.IMAGENET1K_V1)
base_model.classifier = nn.Identity()

class VGG16_with_GAP(nn.Module):
    def __init__(self):
        super(VGG16_with_GAP, self).__init__()
        self.base = base_model.features
        self.gap = nn.AvgPool2d(7)
        self.fc = nn.Linear(512, 10)
        self.softmax = nn.Softmax(dim=1)

        for param in list(self.base.parameters())[:-4]:
            param.requires_grad = False

    def forward(self, x):
        x = self.base(x)
        features = x.clone()
        x = self.gap(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        x = self.softmax(x)
        return x, features
```

Our model



- We will use pre-trained VGG16 for the classification of Cat vs Dogs with image dataset from tensorflow_datasets
- Heatmap can be made by using GAP and multiplying the weights of FC layer and feature maps from the convolutional layer

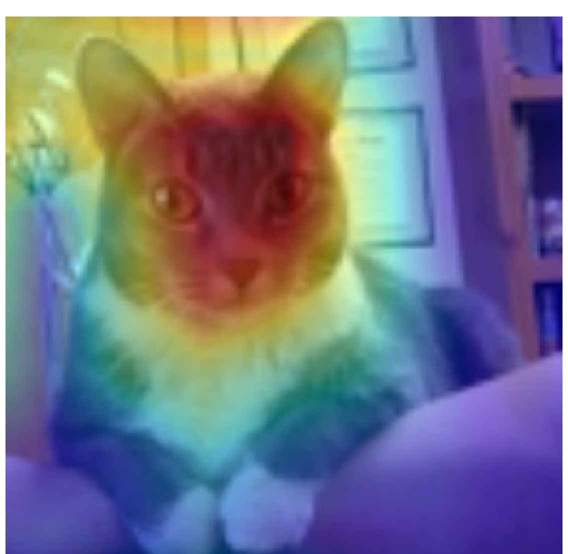
CAM on Cat vs Dogs classification CNN model


```
params = list(model.parameters())[-2]
```

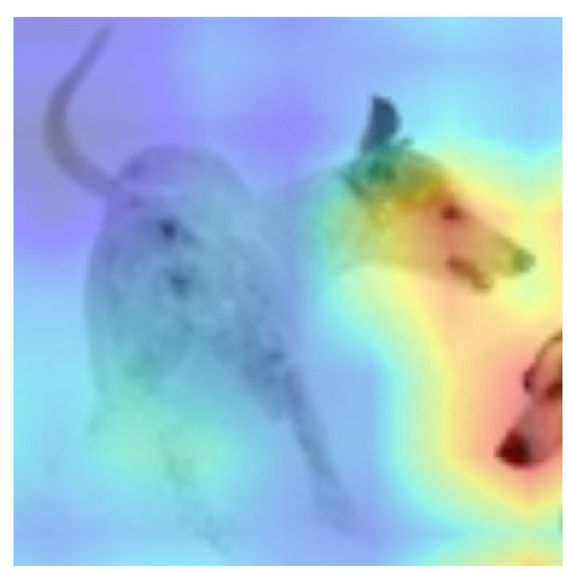
Weight of Dense layer (FC layer)

```
outputs, features = model(images)
overlay = params[int(predicted[num])].matmul(features[num].reshape(512,49)).reshape(7,7)
```

Element-wise product of feature map and weight



Ground Truth : Cat
Prediction : Cat



Ground Truth : Dog
Prediction : Dog