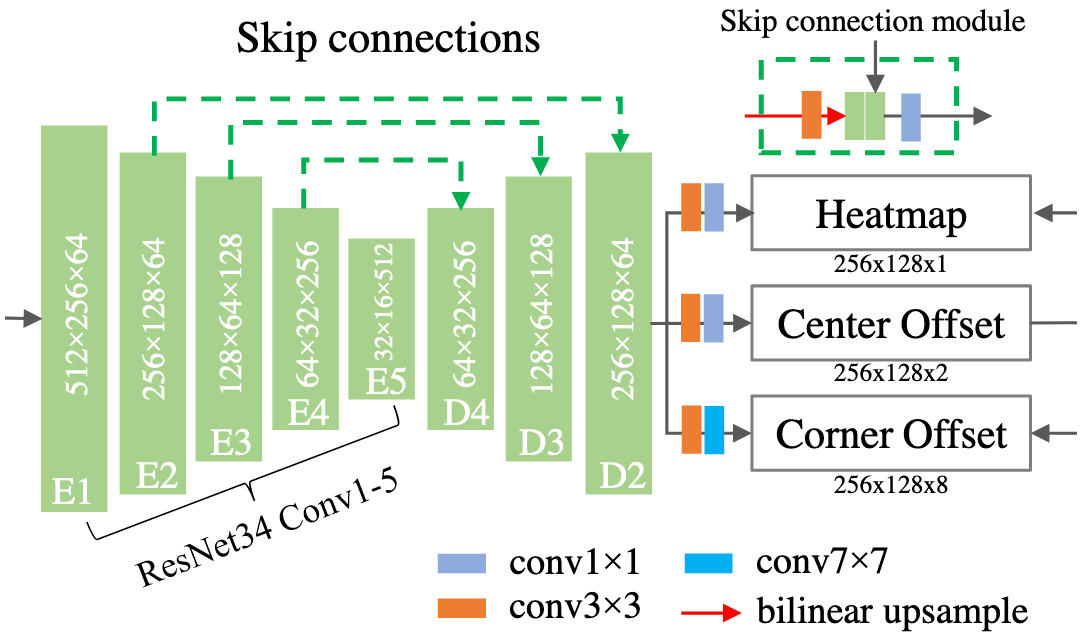
Goal:



ResNet:

(DecNet portion)
The Combination Module Class:

- self.up = nn.Sequential
    - nn.Conv2D(c_low, c_up, kernel_size=3, padding=1, stride=1))
    - nn.BatchNorm2D(c,up)
    - nn.ReLU(inplace=True)
- self.cat_conv = nn.Sequential
    - nn.Conv2D(c_up*2, c_up, kernel_size=1, stride=1)
    - nn.BatchNorm2D(c_up)
    - nn.ReLU(inplace=True)

Decnet: Has 3 dec layers: specified as Combination modules

- self.dec_c2 = CombinationModule(128, 64, batch_norm=True)
- self.dec_c3 = CombinationModule(256, 128, batch_norm=True)
- self.dec_c4 = CombinationModule(512, 256, batch_norm=True)

Network makes some intializations based on "heads". "heads" specified in the Network class of train.py as

heads = {'hm': args.num_classes,

'reg': 2*args.num_classes,
'wh': 2*4,}

The two branches we were looking at have different methods of adding their nn primitives, looks like the image processing one uses the 4dimLayerList, but patrick's branch is just using datapairs and is registering primitives in the traditional way. Unless I am misunderstanding it. I think we should be basing it off of Patrick's branch but was wondering what you guys thought
-Lucas

Decoder:

- Has overall ctdet_decode function
    - Performs max_pool2d such that only certain values are kept in hmax
    - Extracts topk values from torch tensor and gathers the features
- Decoder seems like a lot of math and general array operations used to extract features from data passed in; probably need to create a new primitive for this
- One input into decoder is the heatmap, not sure what the other two are
- I think the other two are center and corner offsets but not sure