Multi-task Self-Supervised Visual Learning <2017>

Summary:

⇒ Deeper networks work better, and that combining tasks - even via a naive multi head architecture - always improves performance.

Although many pretext-tasks had been introduced, they still perform worse than the networks pre-trained on ImageNet.

4 self-supervision tasks chosen for use:

① Relative-position
② Colorization
③ The "exemplar" task
④ motion segmentation

∴ Multiple tasks work better than one, and explore which combinations give the largest boost.

Why a naive combination of self-supervision tasks might conflict:

Reason #1 : Conflict of input channels ---> get more similar inputs!
Reason #2 : Conflict of learning tasks ---> use a lasso-regularized combination of features!

〈Self-supervised Tasks〉

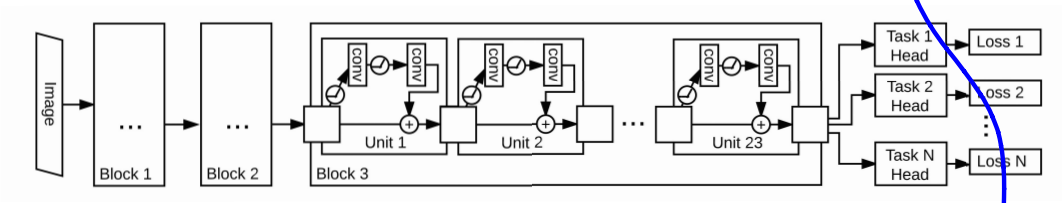
- Relative Position : predict realtive position of patches
- Colorization : predict the color at every pixel
- Exemplar : discriminate between pseudo classes
- Motion Segmentation : classify which pixels will move in subsequent frames.

〈Architectures〉

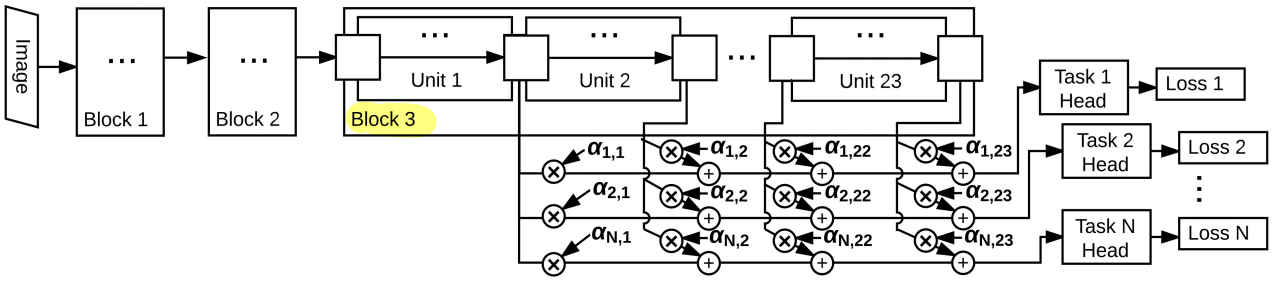
Representation passed to the head for task n:

$$\sum_{m=1}^{M} \alpha_{n,m} * Unit_m$$

① Common Trunk



② Seperating features with Lasso



: a mechanism which allows a network to choose which layers are fed into each task.

skip layer: selects a single layer out of a set of equally sized candidate layers.

→ pass a linear combination of skip layers to each head.

∴ The representation that's fed into each each task head is a sum of the layer activations weighted by these task-specific coefficients.

⊕

Impose a lasso (L1) penalty to encourage the combination to be sparse, which therefore encourages the network to concentrate all of the information required by a single task into a small number of layers.