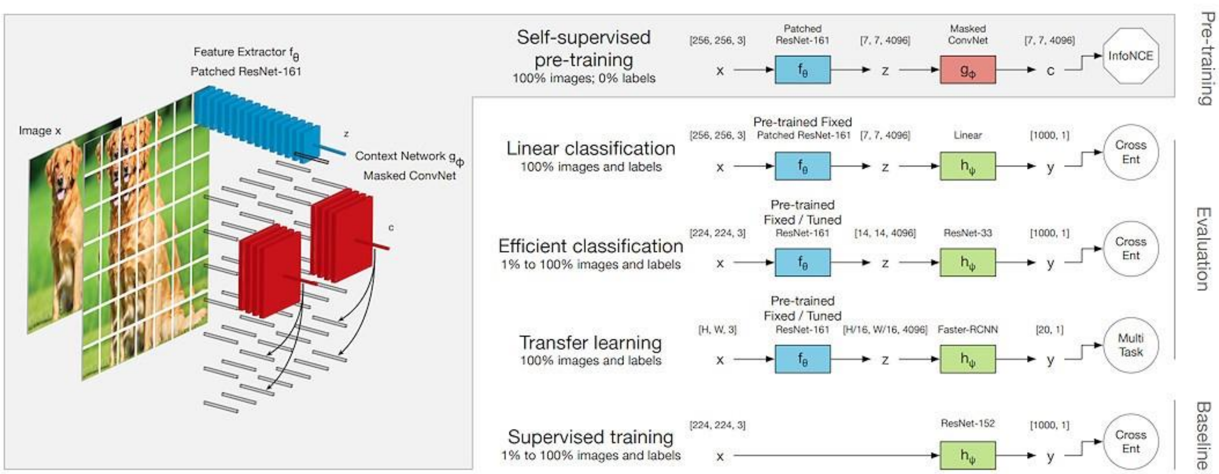
1. **Reducing your labeled data requirements using CPCv2:**

Utilizing CPC 2.0, Image classification and recognition NN's are able to better build representations that allow strong generalization after training on only small amounts of data…getting closer to how humans are able to perform.
Reason: Learns spatial representations in a better way.

Steps:

1) Take large patches of data and pass it through a large model something like ResNet161 and get the features for all the patches.
2) Combine the features from local groups and pass it through the masked ConvNets and using self-supervised learning let your model learn all the feature representations.
3) Replace the Masked ConvNet with a liner layer and train on lsmall labeled data you have.
4) Since, the model has learned better representations we hope to get better results.



Choosing an appropriate training objective is an open problem, but a promising guiding principle has emerged recently: good representations should make the spatio-temporal variability in natural signals more predictable.

The two papers speak about CPC and CPCv2 wherein the first tries to cover its applications across different ML domains extending from videos, audio images to text, whereas CPCv2 is concerned with more of image related matters. The later works better by using different augmentations and larger architectures. The paper also compares its results with the Label propagation techniques(Pseudo-labeling and others) and compares the results with the same.

Major CPCv2 improvements from CPC:
Data augmentation (PA above) — the authors found that by dropping 2 out of 3 color channels on the images served as an excellent augmentation (+3%). They then improved that further by adding more augmentations such as shearing, rotation, color transforms for another 4.5% gain.
Larger model (MC above) — by moving from a ResNet-101 to ResNet-161 (customized) the added a 5% jump. In addition, by integrating larger patches, they were able to boost results another +2%.

The fact that if we good pretrained weights we might not require this going ahead. This technique is mostly concerned with learning good spatial-temporal representations which are used as pretrained weights for the linear classifier.

2. MAML:

1) Meta-learner- the agent teaches the learner or the model.
2) The goal of few-shot meta-learning is to train a model that can quickly adapt to a new task using only a few data points and training iterations.

The objective of this approach is to learn an internal feature that is broadly applicable to all tasks in a task distribution *p(T)*, rather than a single task. This is achieved by minimizing the total loss across tasks sampled from the task distribution *p(T)*
https://pubs.rsna.org/doi/pdf/10.1148/radiol.2019191225

3. Pseudo Labeling:

This technique asks us to build base models and then use these models to predict on unlabelled data we have. Over a certain threshold, we could classify that data into desired classes and do this process iteratively. This process obviously won't help us get 100% correct labels but will hopefully help us get better labels than the current NLP extractor labels.

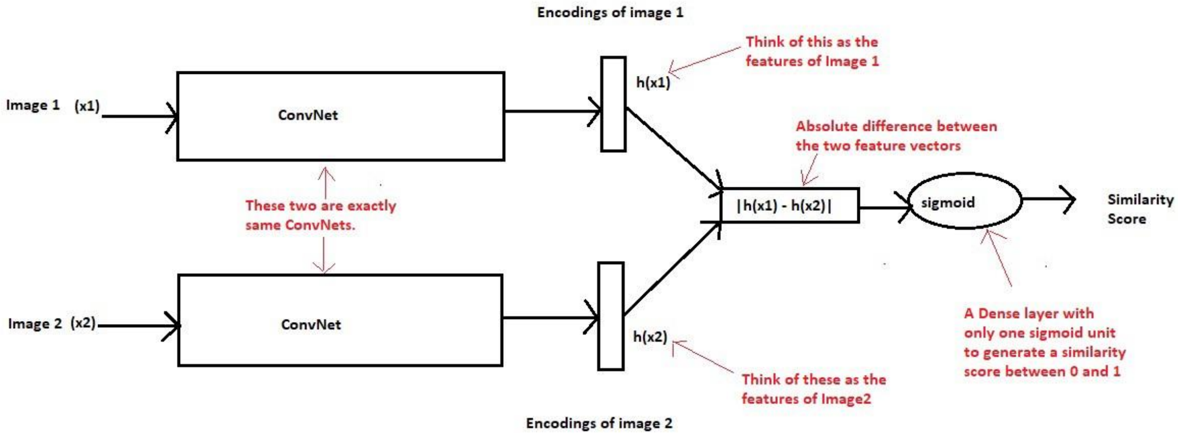
4. AutoEncoders and Other Augmentations:

We haven't been using the power of Augmentations as of now. We have been limited to some augmentations while training but one thing could be

exploiting the power by using as many augmentations. We could try anything ranging from GaussianNoise, Contrast and Sharpness + Affine transformations.
On the other hand, we could use Autoencoders to generate synthetic images, as we know AutoEncoders do not produce the exact same image and have some distortions that could go a long way in taking our models on the path of robustness.

5. Rare Event Classification:
So, this blog basically talks about classifying the less common class in the dataset. What it says, you can treat the less common class as an outlier. Now, take the Normal class samples and train your autoencoder on it. Once that is done, have an approximate value of Reconstruction Error, which is basically equivalent to calculating the loss function while training the model. Now, pass all the samples in your dataset through the trained autoencoder and threshold the value for reconstruction error and on the basis of it decide the outliers.

6. Siamese Networks:
The Siamese network is basically a twin network which compares two images belonging to the same class. The intuition behind this is that the features from two different classes of images will have a significant difference between them. So, basically, given two images the Siamese Network tries to see if they are similar or not.



7. Multi-task learning:

In this you train the model to learn multiple tasks at once. This is different from what we do with multiclass classifiers. In MTL we train the model to learn different tasks and not to differentiate between classes. Here, every dataset could be taken as a data sample. Or consider you have a dataset of Cats, Cars and Birds, Flowers and now you have this third dataset of Dogs with less samples. Now what you want your model to do is to learn the first two tasks efficiently and then use less samples from the third tasks to perform at least equivalent to what it would perform when it had a lot of samples instead. Few shot classification is an instance of MTL.