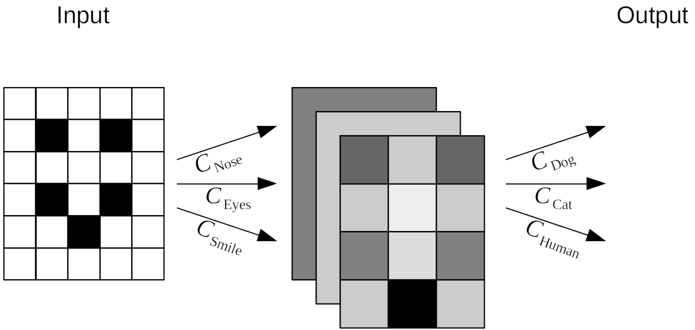
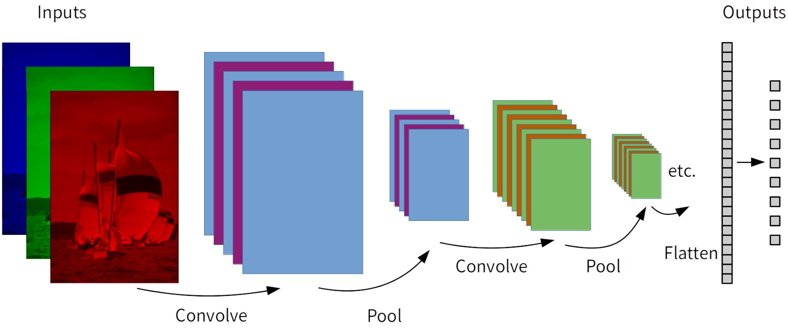
Convolutional Neural Networks

Convolutional neural networks apply multiple convolutions to detect features on multiple scales.



Architectures involve multiple convolution and pooling layers, then *flattening* to treat each pixel separately.



Various strategies are used to prevent overfitting while speeding model training.

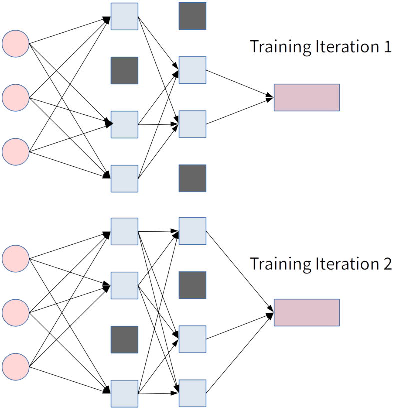
- *Slow learning* using stochastic gradient descent.
  - Multiple iterations through the data set ("epochs") are used to train the model.
  - Weights are tuned on a subset of images at a time.
- *Regularization* penalties are used to ensure weights don't grow to fit only one or two data points.
- *Data augmentation* trains the model on multiple image variants.
- *Dropout learning* randomly removes some nodes from the network in each training step to prevent node overspecialization.

Data augmentation trains the model on multiple versions of the same image.

- The same object won't always be in the same position in an image.
- Rotation, resizing, mirroring, cropping and skewing are all fair game.
- This is a form of *regularization* to prevent overfitting.
- Because training is done in batches, augmentation can be done "on the fly."

Dropout learning, inspired by random forests, forces some nodes to “fill in” for others each training run.

- On training iteration, some nodes are randomly removed.
- In practice, activation functions temporarily set to zero.
- No one node can become overspecialized.



Weight freezing, or using pre-computed layers, speeds model training and reduces data requirements.

- Training image recognition or text analytics requires a *large* amount of data and lots of time.
- Specific applications can build on the same lower-level networks.
- Train only the last few layers for a specific task.
- Examples:
  - `resnet50` for images
  - `word2vec` or `GloVe` for text.