**Summary of the validation accuracies of all the models I tried:**

conv net from scratch: 72.6%

conv net from scratch along with Data Augmentation and Dropout(0.5):82%

conv net from scratch along with Data Augmentation and Dropout(0.5) and extra 2000 more data points: 83.1%

conv net from scratch along with Data Augmentation and Dropout(0.5) and extra 4000 more data points:84.1%

conv net from scratch along with Data Augmentation and Dropout(0.5) and extra 6000 more data points:83.6%

VGG16 pre-trained conv net with just dropout: 90%

VGG16 pre-trained conv net with just dropout with 2000 more data points: 91.7%

VGG16 pre-trained conv net with just dropout with 4000 more data points: 91.9%

VGG16 pre-trained conv net with just dropout with 6000 more data points: 91.9%

Best model from the above is one of the VGG16 models with 6000 training data points and 8000 training data points. I would use the model trained on 6000 training data points and probably tune the hyperparameters like:

1. Filters
2. Number of layers
3. Dropout grid search
4. Different max pooling configuration
5. Fine tuning the models.

Fine-tuning could not be done on my laptop as it does not have GPU.

**Sample size:** It seems like the as the sample size increases, the performance increases up to an extent and it stays flat from a certain sample size number. For the conv net trained from scratch, 6000 data points remained as the best performer and the same with the pre-trained model too.

**Choice of the model:** pre-trained models beat the conv net trained on 8000 data points which means that pre-trained models can do well with small amount of data. If we have small image datasets, pre-trained models are the most suitable and training time is super low.

ConvNets for Computer Vision:

Convolutional Neural Networks (ConvNets) are a type of deep learning model particularly well-suited for computer vision tasks. They're designed to automatically and adaptively learn features from image data, making them highly effective for tasks like object recognition.

Training on Small Datasets:

Even when you have a relatively small dataset, ConvNets can still provide good results. This is in contrast to some other machine learning models that may struggle with limited data.

Overfitting on Small Datasets:

Overfitting is a common problem when working with small datasets. This means the model learns the training data too well and performs poorly on unseen data.

Data Augmentation:

Data augmentation is a technique used to artificially increase the size of your dataset. It involves applying random transformations (e.g., rotations, flips, zooms) to the existing images. This helps expose the model to a wider variety of training examples and can reduce overfitting.

Feature Extraction:

Feature extraction involves using a pre-trained ConvNet as a fixed feature extractor. The early layers of ConvNets learn low-level features like edges and textures, which are generally useful across a wide range of tasks. By using these pre-trained layers and adding some additional layers on top, you can adapt the network to your specific problem without having to train it from scratch.

Fine-Tuning:

Fine-tuning takes feature extraction a step further. Instead of using the pre-trained layers as fixed features, you allow them to be updated during training on your specific dataset. This way, the model can adapt some of the higher-level representations to better suit your task.

In summary, ConvNets are powerful tools for working with image data, and they can yield good results even on small datasets. Overfitting is a common concern, but techniques like data augmentation, feature extraction, and fine-tuning can be employed to mitigate this issue and improve model performance.