**Cross-validation** is when you reserve part of your data to use in evaluating your model. There are different cross-validation methods. The simplest conceptually is to just take 70% (just making up a number here, it doesn't have to be 70%) of your data and use that for training, and then use the remaining 30% of the data to evaluate the model's performance. The reason you need different data for training and evaluating the model is to protect against overfitting. There are other (slightly more involved) cross-validation techniques, of course, like k-fold cross-validation, which often used in practice.

The main steps you need to perform to do cross-validation are:

1. Split the whole dataset in training and test datasets (e.g. 80% of the whole dataset is the training dataset and the remaining 20% is the test dataset)
2. Train the model using the training dataset
3. Test your model on the test dataset. If your model performs well on the test dataset, continue the training process

There are other cross-validation methods, for example

* Leave-one-out cross-validation (LOOCV)
* K-fold cross-validation
* Stratified K-fold cross-validation
* Adversarial cross-validation strategies (used when train and rest datasets are differ largely from each other).

**Grid search** is a method to perform hyper-parameter optimisation, that is, it is a method to find the best combination of hyper-parameters (an example of an hyper-parameter is the learning rate of the optimiser), for a given model (e.g. a CNN) and test dataset. In this scenario, you have several models, each with a different combination of hyper-parameters. Each of these combinations of parameters, which correspond to a single model, can be said to lie on a point of a "grid". The goal is then to train each of these models and evaluate them e.g. using cross-validation. You then select the one that performed best.

To give a concrete example, if you're using a support vector machine, you could use different values for gamma and C. So, for example, you could have a grid with the following values for (gamma, C): (1, 1), (0.1, 1), (1, 10), (0.1, 10). It's a grid because it's like a product of [1, 0.1] for gamma and [1, 10] for C. Grid-search would basically train a SVM for each of these four pair of (gamma, C) values, then evaluate it using cross-validation, and select the one that did best.