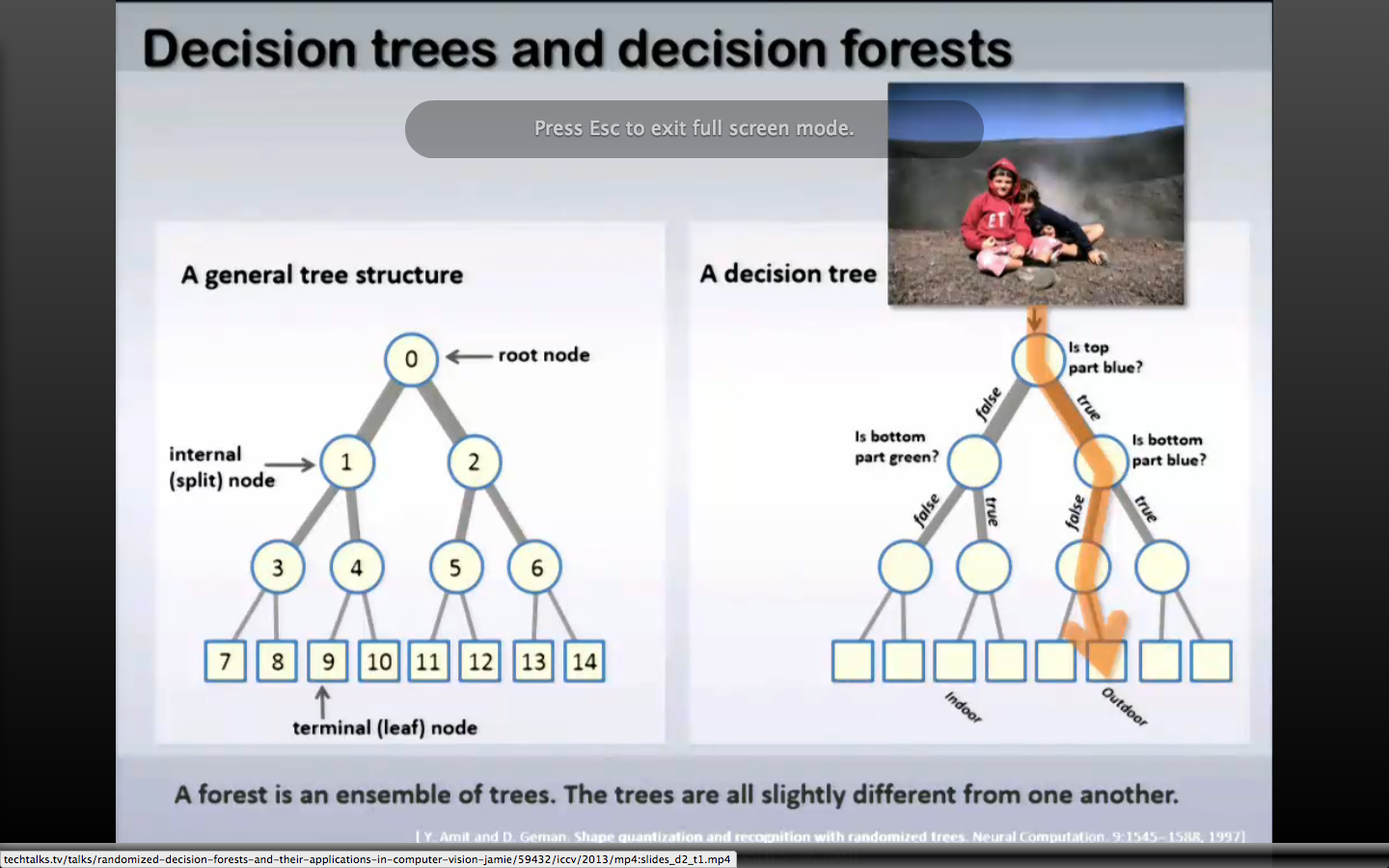
All notes are from the following videos - <http://techtalks.tv/events/320/775/>

## Lecture 1 - Randomized Decision Forests and their Applications in Computer Vision

Start off with a “general tree structure” composed of

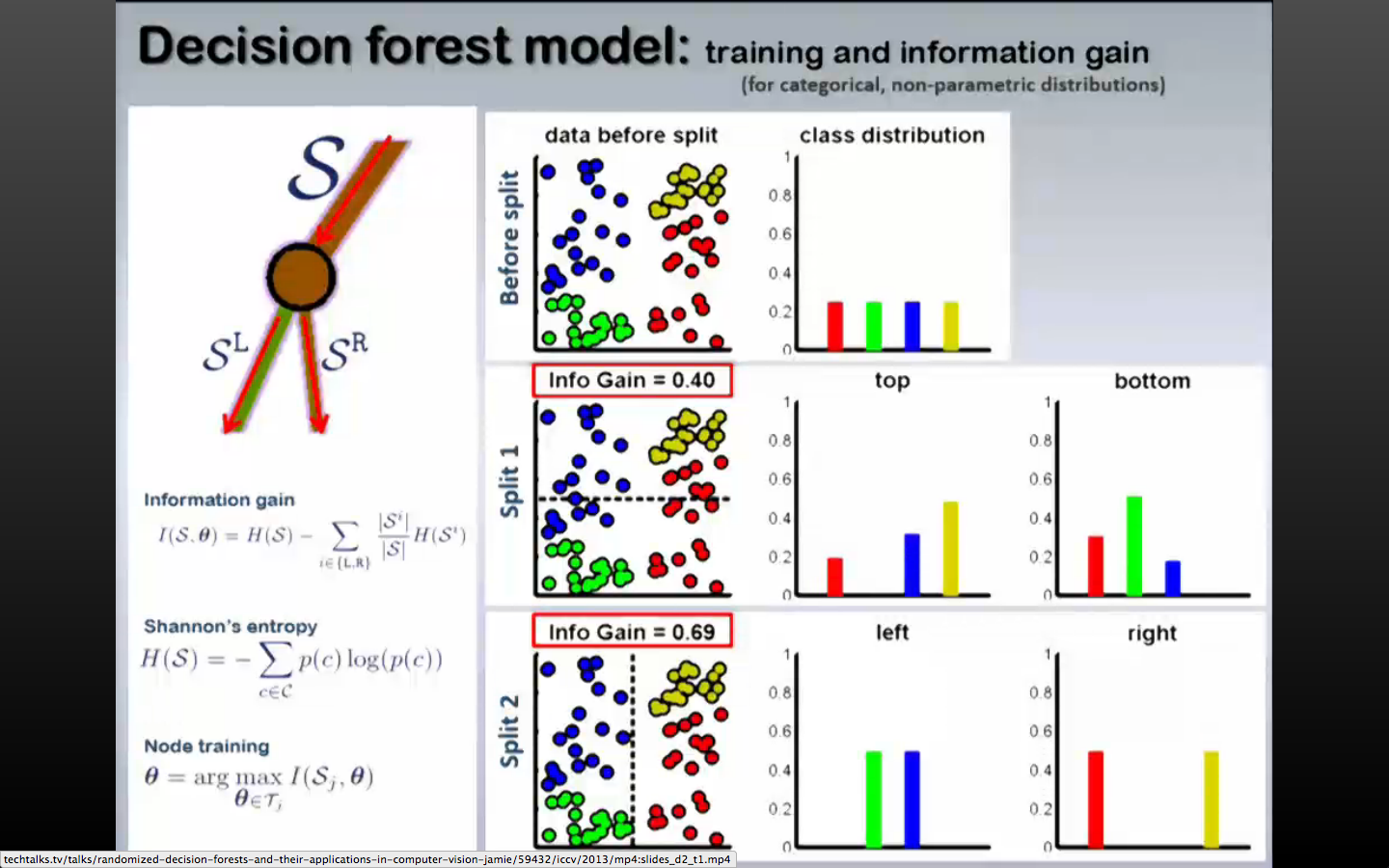
* Root node
* Internal “split” nodes
* Terminal “leaf” nodes



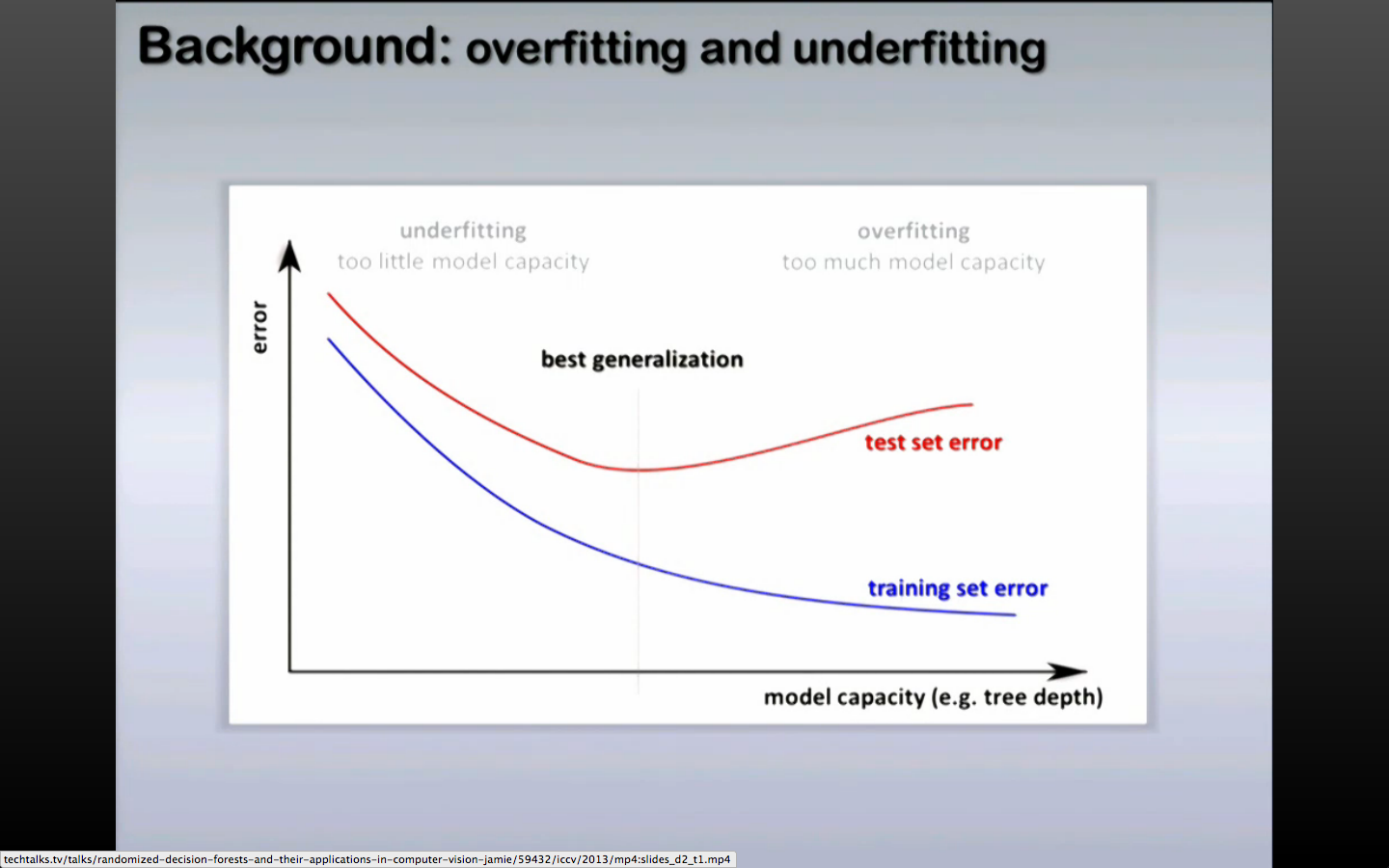
Decision tree: Root node asks a question, and as you navigate to subsequent internal nodes you ask subsequent yes/no questions, and eventually arrive at a terminal node which has a prediction/gives an answer.

Decision forest: Collection of decision trees all trained to answer the same question, but perhaps trained on different data, with different probabilities/weightings, etc.

If you pass a piece of data to multiple trees in the forest, each will produce a (perhaps unique) prediction. How do you combine them into a meaningful result? No straightforward answer, it’s simply a choice you need to be conscious of.



The choice of how you train internal nodes matters, i.e. how you split the data. Sometimes it’s not obvious.



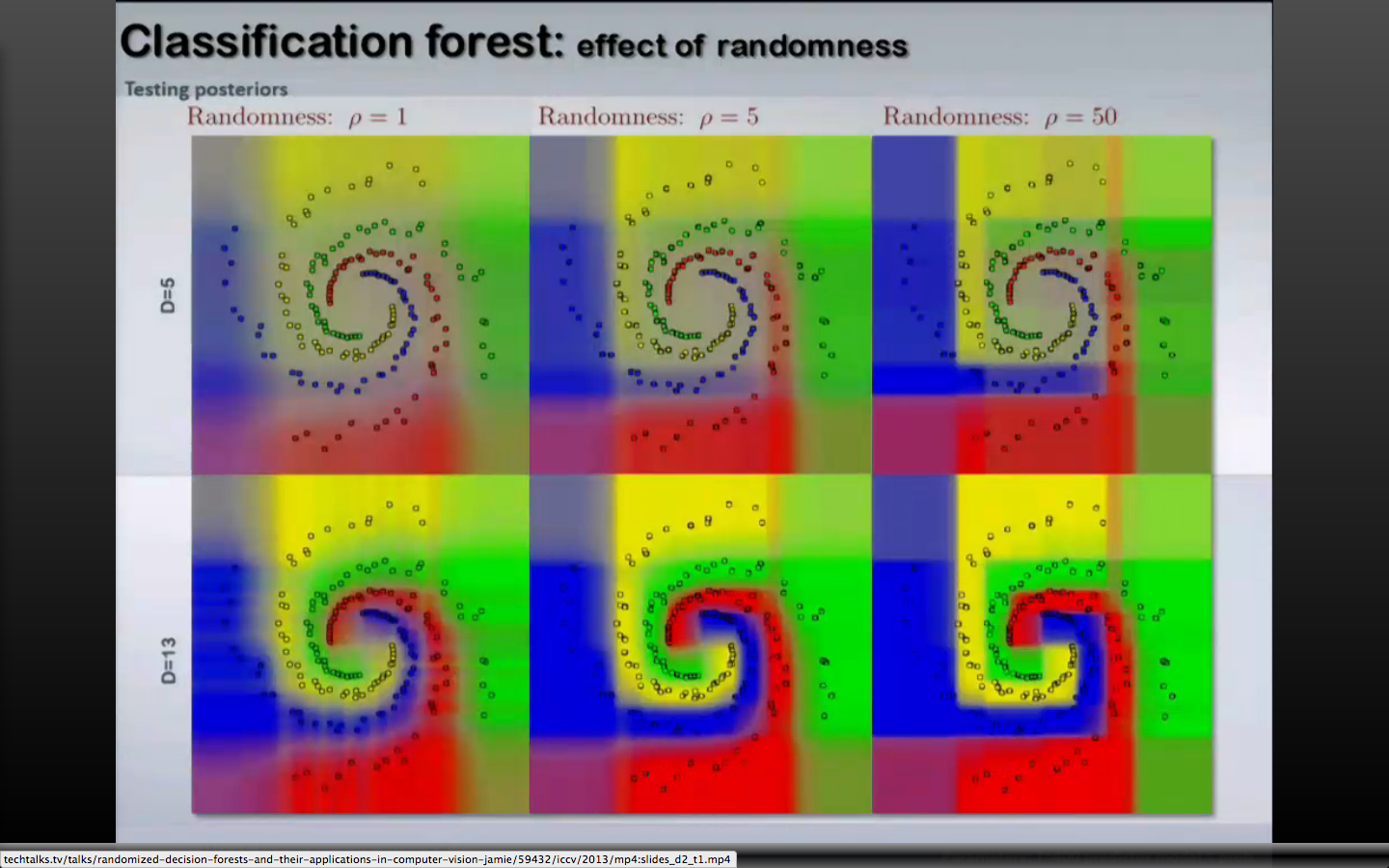
The training set (which you train your model on, comes first) error will go down with an increase in the number of trees in the forest and the length of the forest, since at some point you are just fitting noise (consider the extreme example of one tree per data point). There is a sweet spot where each tree has been trained on a significant amount of data.

* Length of the decision tree is crucial to get right.

Classification schemes for decision trees:

* Weak Learner: drawing simple boundaries to separate the data. Axis aligned, oriented line, conic sections.

“rho” parameter: specifies the number of parameters that you are evaluating at each node when building the tree. Associated with the degree of randomness in the forest. The larger rho is, the more similar each tree in the forest will be.



In figure (using axis aligned weak learners to categorize regions): D = depth of each tree, rho = degree of randomness with each tree (lower number is more random).

* For rho=1 (high randomness), low tree depth does not give good classification boundaries (very light colours, not well defined boundaries), but for a deep tree, it’s actually pretty good.
* For rho=50 (less randomness), low tree depth gives much better results than before, while for high tree depth it is starting to overfit.

Terms:

Bagging: The same as cross validation – essentially split your training data into subsets and train each tree using these subsets. Helps prevent overfitting.