KNOWLEDGE DISTILLATION FROM RANDOM FOREST

Clide Dcosta

***Abstract*—Over the years there has been a number of significant works that combine two more classifiers to produce a single classifier. The resulting classifier, is a set of classifiers whose individual decisions are combined into one predictive model in order improve prediction. The accuracy predicted from ensemble classifiers are more accurate than of the individual classifiers that make them up. However, making predictions using an ensemble method is cumbersome. This paper suggests a method to compress the knowledge in an ensemble into a single model which is easier to deploy. Random forest being an ensemble classifier has very high predictive power. To improve on its prediction, this paper suggests a technique where the probabilities are extracted from the model and trained against a decision tree classifier. In theory the knowledge encoded within a collection of trees is compressed into a single interpretable tree.**

I. Introduction

Recent advancements in machine learning algorithms have been nothing short of phenomenal. Better models are being built that give better accuracy and are better performing as the years go by. This has been possible due to a number of theoretic and algorithmic breakthroughs in the last decade or so, facilitated by significant technological advancements in computer hardware and software [1]. Large Ensembles which are a combination of many different individual classifiers are an example of a particular trend that has been introduced in recent years. One famous ensemble method is Random forest classifier.

Random forest is an algorithm for classification that uses an ensemble of classification trees. Each of the classification trees is built using a bootstrap sample of the data. At each split the candidate set of variables is a random set of variables. The random forest classifier uses bagging(bootstrap aggregation) and Random variable selection for tree building. Each tree is grown fully so as to obtain trees with low bias and low variance. The bagging and random feature selection techniques are used to produce trees in which the correlation is minimum. The advantage of Random forest classifiers over other classifiers is that it runs efficiently on large databases and it produces the most accurate predictions when compared to other classifiers.

Random forest being an ensemble method also results in an increase in model size and complexity. As a result of its large size and complexity, a number of significant practical challenges arise [1]. Random forest is a collection of decision trees that have several hidden layers which results in evaluation time of model to increase especially if the n_estimators (number of decision trees) are large. Due to its large size and sophisticated structure, Interpretability will reduce and integration into larger systems will be hard. Size and complexity may present models with power, but they can also make them cumbersome and difficult to use [1].

Cumbersome models gain knowledge from a large number of classes to discriminate. The normal training objective is to maximize the average log probability of the correct answer and assign probability to all the classes, with some classes given small probabilities with respect to others. The relative outcomes of incorrect answers say a lot regarding the generalization of this complex model. An image of a Car, for example, may only have a very small chance of being mistaken for a Truck, but that mistake is still many times more probable than actually predicting it's a car. To overcome the liabilities of the cumbersome models,we have to transfer the knowledge of these models to a smaller model which can be interpreted easily . We can build another model from the cumbersome model that meets performance criteria in terms of accuracy and efficiency but is no longer cumbersome and is found to be convenient. Here, a small model is trained to mimic a pre-trained larger model(ensemble of models).

We introduce a framework for building such convenient models from cumbersome models. This procedure is called knowledge distillation; as though the knowledge hidden inside the cumbersome model could be extracted, distilled and injected into the convenient model. The knowledge is exchanged from the cumbersome model to a more refined model i via preparing it in the exchange set and utilizing a soft target distribution for each case in the exchange set that is created by utilizing the bulky model with high temperature in the softmax. A framework is introduced for building such convenient models from cumbersome models. To put this into objective, a random forest model will be trained against the data and their probabilities will be used to create a new dataset. The concept of knowledge distillation will be applied whereas a decision tree will be trained on the new dataset and their accuracy, recall, precision and F1 scores will be compared against the original dataset.

II. Background

In ensemble classification multiple classifiers are used to predict the model and the accuracy generated is significantly much higher than that of the individual classifiers. A voting technique is used to determine the class label for the variable being considered. A simple but effective voting scheme is majority voting[2]. In majority voting each classifier in the ensemble is asked to predict the class of the unlabeled instance. Once all the votes have been tallied up, the classifiers having the maximum number of votes is returned as the predicted class of that unlabeled instance. Another alternative voting scheme used is called Veto voting. Here one single classifiers vetoes the decision of other classifiers[3]. The most common

voting scheme that recently burst into existence was trust based veto voting [4]which is an extension of veto voting. The trust of each classifier in the ensemble is determined and used to find out whether a classifier or set of classifiers can veto the decision. In ensemble classification various classifiers are utilized to predict the model and the accuracy created by the model is a lot higher than individual classifiers.The objective is to find out the process where the cost of computation is low but gives higher accuracy.This idea in theory can be termed as knowledge distillation.

The random forest algorithm is a case of packing classifier where several decision trees is packed to form an ensemble of trees in a randomized forest.Breiman [9] was the first to provide a general definition of a Random forest and also proposed what is arguably the most popular Random forest algorithm. Concretely, Breiman defines an RF as follows:

> A random forest is a classifier consisting of a collection of tree-structured classifiers {t(x, Θb), b = 1, ..., B} where the {Θb} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.
>
> *[9]*

Once the cumbersome model is prepared, to instill the concept of knowledge distillation, a substitution method is used where in the information from a complex model is structured and transferred into a more straight forward model that is more suitable for deployment and can be interpreted easily. Knowledge distillation was introduced only recently by[5], who used it to mean model compression. Knowledge distillation is effective to train the small and generalisable network models for meeting the low-memory and fast running requirements. Knowledge distillation compacts deep networks by letting small student network learn from a large teacher network[6]. Knowledge distillation exchanges learning from a pre-prepared vast "instructor" network to a little "student" network, for encouraging the deployment at test time. Initially ,this is finished by relapsing the softmax yield of the instructor model [5].

The knowledge distillation for the purpose of model compression was first proposed by[7]. A related work [8] explores the relationship between depth and width of a neural network in terms of capacity and representation power.This involves training a wide but shallow student network with deep teacher network. Hinton et al. finally generalizes and re-formulates knowledge distillation in[5], where in a student is set to utilize the information contained in the "soft targets" from teacher's softmax probability distribution(softened by temperature hyper parameter) to aid the training of student models for the same task. These soft targets may have dark knowledge helpful for student models to also learn from what is hidden in the incorrect classes that a trained teacher produces given the same sample.

III. Methodology

A complete analysis on the given datasets will help to give a clear insight into the various attributes present in the data. Initially the dataset and the necessary libraries will loaded into the sci-kit learn environment . . For the most part, real-world data is incomplete, inconsistent, lacking in certain behaviors or trends, containing outliers, noise, aggregate data, and prone to many errors. Preprocessing data is a proven way to solve such problems. The initial pre-processing stage will involve a small data visualization of all the features present within them. Missing values are then checked and imputed with its relevant mean, mode or median . Outliers are also detected and handled with.Once the intial stage of pre-processing is completed, The next task is to normalize the features present within the dataset. Normalization can be defined as a technique often applied as part of data preparation for machine learning. The objective of normalization is to change numeric column values in the dataset to use a common scale without distorting differences in value ranges or losing information. For some algorithms, standardization is also required to properly model the data.

To ensure that there's a fair distribution of classes in the training data. Splitting and sampling helps to make sure that additional data is not getting processed. For this particular case I have chosen to use a validation set approach where in the data is split into train and test data according to 70:30 split respectively. The data is split with the help of the train_test_split functionality provided by sklearn.

Since random forest being an ensemble classifier which gives high accuracy and low bias and variance, I have chosen this as the base model through which i can distill its knowledge to another shorter classifier (Decision tree). Once the model has been trained, the prediction accuracy along with is precision, recall and F1 scores are calculated . The next step involves generating the probabilities associated with the model based on the target variable. The function predict_proba is used to retrieve these probabilities. The RandomForest simply votes among the results. predict_proba() returns the number of votes for each class (each tree in the forest makes its own decision and chooses exactly one class), divided by the number of trees in the forest. Hence, your precision is exactly 1/n_estimators.

The predictions made by the predictive model can be calibrated.Calibration of prediction probabilities is a rescaling operation that is applied after the predictions have been made by a predictive model.To asses the calibration of the model,i have chosen to visualize calibration plots.Once predictions are chosen for the number of probability bins in the bin, each bin will then be converted to a point on the plot. The y-value for each bin is the proportion of true results, and the x-value is the average probability predicted. A well-calibrated model therefore has a calibration curve hugging the line y=x straight. Here is an example of a two-curve calibration plot, each with a model on the same data .

A new dataset will be created by inducing the probabilities generated from the previous dataset.This dataset will be classified on a multi-class level and then the knowledge will be distilled. The new dataset will be trained with the decision tree algorithm (distillation model)to predict their accuracy before and after distillation procedure.This performance is also compared against other classification algorithms to determine

the level of succes. The various libraries used for this task includes Numpy,Pandas,Matplotlib etc..

The three datasets selected for this project are ;The first data set is the heart disease data. The heart disease data set has 12 predictor variables and one target variable which has values 1 and 0 . The target variable depicts whether the person has heart disease (1) or not(2).The total number of instances present here is 305

The second data set is the red-wine quality dataset.The red wine quality data set has a total of 13 variables i.e. 12 predictor variables and one target variable which is quality.The target variable has a range of values ranging from 0 to 10 with zero being the wine with the worse quality and 10 being the one with the highest quality.The number of instances present here is 1.7k.

The third dataset is the German credit risk dataset .This dataset contains 10 attributes. Each entry represents a person who takes a credit from a bank The response variable is the Risk in which each person is classified as good or bad credit risks.The number of instances present here is 1000.These datasets in questions can be found on the UCI repository.

Let us look at few terminologies mentioned here.

A. Knowledge Distillation

The main concept behind Knowledge Distillation is to transfer the knowledge of the cumbersome model to a small model which can be easily exported.Knowledge distillation is model compression method in which a small model is trained to mimic a pre-trained, larger model. This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student. We can consider the cumbersome model as Teacher Network and our new small model as **Student Network. In** distillation, knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model. However, in many cases, this probability distribution has the correct class at a very high probability, with all other class probabilities very close to 0. As such, it doesn't provide much information beyond the ground truth labels already provided in the dataset. To tackle this issue, Hinton et al., introduced the concept of "softmax temperature"[5]. The probability p_i of class i is calculated from the logits as:

$$p_i = exp(z_i/t)/P_j\ exp(z_j/t)$$

where t is the temperature parameter. When $t = 1$ we get the standard softmax function. As T grows, the probability distribution generated by the softmax function becomes softer, providing more information as to which classes the teacher found more similar to the predicted class. When computing the loss function vs. the teacher's soft targets, we use the same value of t to compute the softmax on the student's logits. We call this loss the "distillation loss". The overall loss function, incorporating both distillation and student losses, is calculated as:

$$L(x;W) = \alpha * H(y,\sigma(z_s;T = 1)) + \beta * H(\sigma(z_t;T = \tau)$$

where x is the input, $W$are the student model parameters, $y$is the ground truth label, H is the cross-entropy loss function, $\sigma$is the softmax function parameterized by the temperature T, and $\alpha$and βd are coefficients. z_s and $z_t$are the logits of the student and teacher respectively.

Soft targets provide much more information per training case than hard targets when they have high entropy and less variance in the gradient between training cases, so that the small model can often be trained on much less data than the original cumbersome model while using a much higher learning rate. Much of the learned function information lies in the very low probability ratios in the soft targets. This is valuable information defining a rich structure of similarity over the data.

B. Model Compression

Increasing the capacity of a single model or combining multiple models into an ensemble traditionally improves predictions; but it does comes at a cost. The final model might be too big to store, or too slow to evaluate at the time of testing.

Given a discriminative model which computes the function$y = t(x)$,this model is referred as the "teacher". We then specify a family of models$f(x|\theta)$, parameterized by a set of parametersθ, which we shall refer to as the "student". Our goal is to train the student to be as similar as possible to the teacher; that is, we want to find the value of θ for which the approximation

$t(x) \approx f(x|\theta)$

In the context of model compression, $t(x)$ is typically significantly more complex than$f(x|\theta)$, therefore, in general, we cannot expect the student to match the teacher everywhere. The hope instead is that the particular function that is represented by $t(x)$ is simple enough to be closely approximated by $f(x|\theta)$ at least in the region of space we are interested in.The framework built here for the model compression, besides the teacher$t(x)$ and the student$f(x|\theta)$, also involves a probability generator. The probability generator is a probability distribution over the input space; in practice it can be any procedure which can generate datapoints on demand. These datapoint represents the probability bins for the target variables indicating the probability distribution that each target variable can belong to different classes .

C. Decision Tree Algorithm

Decision trees are individual learners that are combined. They are one of the most popular learning methods commonly used for data exploration. One type of decision tree is called CART...classification and regression tree. CART ...greedy, top-down binary, recursive partitioning, that divides feature space into sets of disjoint rectangular regions. A decision tree can be used for both classification (ie a classification tree) or regression (ie regression tree) tasks. Classification trees are grown to classify objects according to their specific attribute value(ie the particular class the variable belongs to). Their most common applications can be found in the field of finance or medicine. Classification trees give a hierarchical decomposition of the data. Regression tree can be classified under the bracket of

decision tree where the target variable is continuous. They attempt to predict the values of the continuous variable from one or more continuous and/or categorical predictor variable.

D. Random Forest Algorithm

Random forests is an ensemble classifier that consists of many decision trees as input and outputs the class that is the node of the class's output by individual trees. The method combines Breiman's bagging idea and the random selection of features. In Random forest we analyze the data by first selecting a target attribute. In applications of business this target value may be the final status of a loan or credit card account, for detection of fraud status of insurance claim.

The Random forest algorithm works in the following way:

1. Random Subset Selection: Each tree is trained on roughly 2/3rd of the total training data . Subsets are drawn at random with replacement from the original data. This sample will be the training set for growing the tree. **2. Random Variable Selection:** Some predictor variables say n are selected at random out of all the predictor variables and the best split on these n is used to split the node.

3. For each tree, the data not used for training is used to calculate the misclassification rate also known as the out of bag (OOB) error rate. If we grow 500 trees then on average a record will be OOB for about .37*500=185 trees.37 refers to the percentage of data not used for training.

4. Each tree in the random forest model presides a classification on the data which is not used for training.(OOB). The tree votes for that class. The forest chooses the classification having the most votes over all the trees in the forest. For a binary dependent variable, the vote will be 1 or 0, count up the 1 votes. This is the RF score and the percent 1 votes received is the predicted probability. In regression case, it is average of dependent variable.

IV. Experiments

The three datasets chosen form the basis of a classification task. Each target variable present within the dataset belongs to a certain class. The procedure followed in the experiment follows a similar pattern for all the three datasets.

All the variables within the dataset are normalized so that the distribution among the variables follow a Gaussian distribution. A random forest is trained on the datasets with a reasonable high number of trees. The probabilities of each class within the dataset is calculated and a new dataset is included that contains these probabilities. Decision tree classifiers (Distillation model) are then trained on the new datasets and their performance is compared with the original dataset. Other state-of-the-art machine learning algorithms like K-NN, Naive Bayes, SVM and Logistic regression are trained on both the original and new dataset and the performance metrics are compared.

A. Heart disease dataset

The target variable predicts whether the person has a heart disease or not. A small data analysis on the data gives a little insight to the type of data we are dealing with.

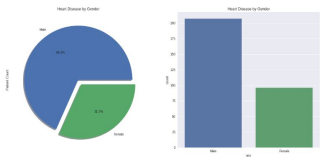

Fig. 1. Heart disease according to gender

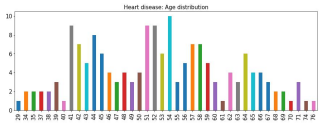

Fig. 2. Distribution of diseases according to ages

Figure (1)shows the distribution of heart diseases among the patients of various genders. We can see that more men have heart issues than women.Figure(2) shows the distribution of diseases among various ages of the patients. It is evident that throughout the ages 51 to 55 , there is a high chance that the person will have a heart disease.

Lets us now move on to the next part of the experiment in this , where modeling will be done.After normalizing all the features within the dataset,A sample and split will occur wherein the data set is splitted to train and test according the ratio 70:30 respectively. A random forest model with number of trees equal to 100 is trained on the data and their probabilities are calculated.

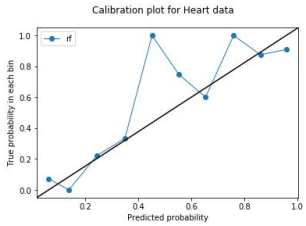

Fig. 3. Calibration plot heart data

Using a calibration plot, I asses the probabilities that are generated by the random forest model. Calibration refers to whether the future predicted probabilities agree with the observed probabilities. A well-calibrated model has a calibration curve that hugs the straight line y=x. . We can see in figure(3) that most of the probabilities are in and around the straight line.

After adding these probabilities into the original data set, a distillation model(Decision tree) is run against it. The performance metrics are than compared between the two datasets. The prediction accuracy obtained when running the random forest model on the original dataset was 81.3%.After adding the probabilities and running the distilled model, the accuracy prediction rose by a staggering 12 percent with a score of 93.08. A decision tree is run against the original dataset and all the three models metrics are compared in table(I)

Other state -of-the art methods like K -Nearest neighbors, SVM,Logistic regression and Naive Bayes algorithms are modeled against both the original dataset and the new dataset.

TABLE I
COMPARISON OF MODEL

Model	Accuracy	Precision	Recall	F1 scores
Random forest	0.813	0.823	0.84	0.8316
Decision tree	0.725	0.777	0.7	0.736
Decision tree (Distilled model)	0.93	0.94	0.944	0.941

Table(II) gives a comparison of accuracy obtained on the original dataset and the distilled model dataset.

TABLE II
COMPARISON OF ALGORITHM ON ORIGINAL AND NEW DATASET(HEART DISEASE)

Algorithms	Original dataset	New dataset
DT	0.72	0.93
LR	0.79	0.98
KNN	0.81	0.93
SVM	0.84	0.98
NB	0.83	0.96

B. Red Wine quality dataset

This dataset is viewed as classification task. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). The attributes associated with this dataset are 1 - fixed acidity 2 - volatile acidity 3 - citric acid 4 - residual sugar 5 - chlorides 6 - free sulfur dioxide 7 - total sulfur dioxide 8 - density 9 - pH 10 - sulphates 11 - alcohol Output variable (based on sensory data): 12 - quality (score between 0 and 10) .

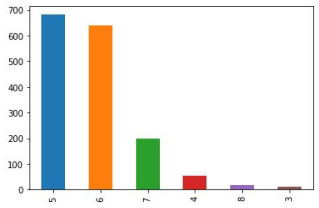

Fig. 4. Distribution of quality(target) variable

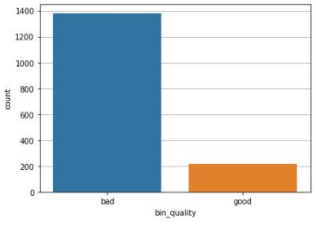

Fig. 5. Redistribution of quality

As we can see in figure(4) the most of the distribution is distributed towards 5 and 6.We assume an arbitrary cutoff for your dependent variable (wine quality) at e.g. 7 or higher getting classified as 'good/1' and the remainder as 'not good/0'. This allows us to practice with hyper parameter tuning on e.g. decision tree algorithms looking at the ROC curve and the AUC value. Figure(5) gives the redistribution of quality variables according to the probability bin set.

After normalizing all the features within the dataset,A sample and split will occur wherein the data set is splitted to train and test according the ratio 70:30 respectively. A random forest model with number of trees equal to 100 is trained on the data and their probabilities are calculated.

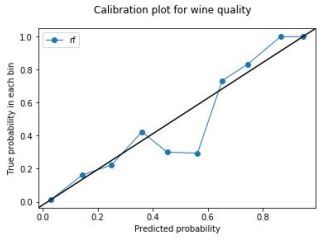

Fig. 6. ROC -AUC curve for red wine quality dataset

Figure(6) shows the AUC-ROC curve. Higher the AUC better the model is at predicting 0s as 0s and 1s as 1s.ROC-AUC of this dataset was found to be close to one, so its prediction accuracy is high.

These probabilities into the original data setaccording to probability bins created for each class present in the target variable. A distillation model(Decision tree) is run against it. The performance metrics are than compared between the two datasets. The prediction accuracy obtained when running the random forest model on the original dataset was 89.1%.After adding the probabilities and running the distilled model, the accuracy prediction rose by a 6 percent with a score of 95.625. A decision tree is run against the original dataset and all the three models metrics are compared in table(III)

TABLE III
COMPARISON OF PERFORMANCE METRICS

Model	Accuracy	Precision	Recall	F1 score
RF	0.89	0.647	0.492	0.599
DT	0.86	0.517	0.656	0.578
DT on RF model	0.95	0.710	0.90	0.79

Other state -of-the art methods like K -Nearest neighbors , SVM ,Logistic regression and Naive Bayes algorithms are modeled against both the original dataset and the new dataset . Table(1V) gives a comparison of accuracy obtained on the original dataset and the distilled model dataset.

C. German credit risk data

The German credit dataset includes 10 predictor variables and one response variable (Risk). Based on the predictor variables , the target variable specify whether a person who has taken a loan from the bank is a credit risk or not. This dataset contains categorical data so it is essential to convert this into numeric data.

TABLE IV
COMPARISON BETWEEN ORIGINAL AND NEW DATASET(WINE QUALITY)

Model	Original dataset	New dataset
DT	0.864	0.956
LR	0.870	0.978
KNN	0.868	0.965
SVM	0.860	0.975
NB	0.835	0.946

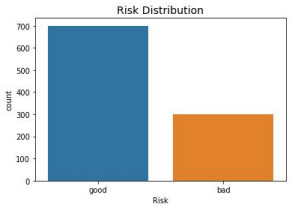

Fig. 7. Distribution of target(Risk) variable

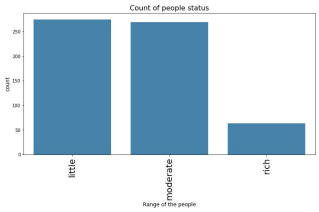

Fig. 8. Checking account of people of various background

From figure(7) we can have an idea about how many people are categorized under credit risks. The ratio between bad and good credit risks can be approximated as 2:1. Figure(8) gives the distribution of checking account based on whether the person has enough funds to be classified as either little, moderate and rich.

In the next part we will use a random forest model to model the data. done.,A sample and split will occur where in the data set is splitted to train and test according the ratio 70:30 respectively. The random forest will be trained with the number of trees equal to 100 is trained on the data and their probabilities are calculated.

A probability bin is created with an assigned set of probabilities . The probabilities generated from the random forest model will then be classified under these bins for each class present in the target variable. Our next step is to consider any one of the classes generated from the bin distribution and run the distillation model(Decision tree) on this new dataset.

The accuracy of the model before the distillation process(Random Forest) was found to be 74.33% and the accuracy afters was recorded as 89 table(5)showing a massive increase by almost 15 %.

Classification algorithms like KNN,Naive Bayes,SVM and logistic regression are also modeled and compared against the original dataset and the new dataset table(V1).

V. CONCLUSION

In this paper we have discussed the concept of knowledge distillation through tree based methods. We have shown that distilling works very well for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model. Using sci-kit learn we have trained random forest model and transferred the information into another model. The knowledge extracted from the cumbersome model is distilled into a more convenient model that can be interpreted easily and computationally less expensive to deploy. The prediction accuracy vastly improve after the model is distilled.The accuracy jump could be explained because of the size of the data used. If the model is trained against a large dataset, The accuracy rise will not be as much.

TABLE V
COMPARISON OF PERFORMANCE

Model	Accuracy	Precision	Recall	F1 score
Random Forest	0.743	0.77	0.89	0.83
Decision Tree	0.89	0.91	0.92	0.923
Decision tree on RF	0.66	0.74	0.784	0.766

TABLE VI
COMPARISON OF ACCURACY BETWEEN ORIGINAL AND NEW DATASET(GERMAN CREDIT)

Model	Original Datset	New dataset
DT	0.66	0.89
LR	0.72	0.895
KNN	0.73	0.89
SVM	0.69	0.895
NB	0.71	0.88

VI. REFERENCES

[1]George Papamakarios,Distilling Model Knowledge(2015) L. Breiman, bagging predictors Machine Learning, vol. 26, pp. 123-140, 1996

[2]Lam, L. and Suen, C.Y. (1997) Application of majority voting to pattern recognition: An analysis of its behavior and performance. IEEE Transactions on Pattern Analysis, 27, 553-568.

[3]Shahzad, R. K., & Lavesson, N. (2012). Veto-based malware detection. In 2012 seventh international conference on availability, reliability and security (ARES), Prague, Czech Republic (pp. 47–54). New York City, NY: IEEE.

[4]Sun,Y.-A.,&Dance,C.(2012).When majority voting fails: Comparing quality assurance methods for noisy human computation environment. CoRR. Retrieved from arXiv:1204.3516.

[5]G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[6]Silvia L. Pintea, Yue Liu, Jan C. van Gemert,Recurrent knowledge distillation,arXiv:1805.07170

[7]J Dai, Y Li, K He, and J Sun, "R-fcn: Object detection via region-based fully convolutional networks," in NIPS, 2016, pp. 379–387.

[8]Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Advances in neural information processing systems, pages 2654–2662, 2014

[9]L. Breiman, Random Forests, 2000.