A Novel One-vs-Rest Classification Framework for Mutually Supported Decisions by Independent Parallel Classifiers

Antonios Vogiatzis
School of ECE
Technical Univ. of Crete
Chania, Greece
anvogiatzis@intelligence.tuc.gr

Georgios Chalkiadakis
School of ECE
Technical Univ. of Crete
Chania, Greece
gehalk@intelligence.tuc.gr

Konstantia Moirogiorgou
School of ECE
Technical Univ. of Crete
Chania, Greece
dina@display.tuc.gr

Michalis Zervakis
School of ECE
Technical Univ. of Crete
Chania, Greece
michalis@display.tuc.gr

***Abstract*—We put forward a generic classification architecture of independent parallel CNNs that explicitly exploits a "mutual exclusivity" or "classifiers' mutually supported decisions" property underlying many dataset domains of interest, namely that in many cases an image in a given dataset might almost unquestionably belong to one class only. Our framework incorporates several designed-to-purpose opinion aggregation decision rules that are triggered when the mutual exclusivity property is or is not satisfied; and makes use of "weights" which intuitively mirror the confidence each CNN has in identifying its corresponding class. Our framework can thus (a) take advantage of clear class boundaries when these exist, and (b) effectively assign items to classes with increased confidence, even when clear class boundaries do not exist. We confirm the effectiveness of our approach via experiments conducted on a well-known dataset from the waste classification domain.**

***Index Terms*—image classification, supervised learning, mutual exclusivity, decision rules, opinion aggregation**

I. Introduction

Many image datasets typically contain images with drastically different characteristics, allowing for the training of different classifiers with each of which succeeding in achieving high confidence levels. Examples include *(a)* recyclable materials: for instance, glass objects are very different from plastic, making it easy to differentiate between the materials; *(b)* facial recognition problems: e.g., the color of the skin and the shape of the eyes are powerful characteristics which facilitate the required separation; and *(c)* X-ray datasets: in such images, the bone and tissue density provide strong signals that can unambiguously denote certain medical conditions.

In this paper we put forward a generic classification framework that is well-suited for such settings. Our approach casts a single label multi-class classification problem into one to be tackled via the use of as many parallel binary classifiers as the different classes. Our framework makes use of a neural networks-based architecture which tackles the multi-class classification problem as one composed by multiple "binary" classification ones. As such, our approach belongs to the *one-vs-rest* family of classification methods. Importantly, our classification framework comes complete with a set of *decision rules* which correspond to social choice functions (or "voting rules") [1] that enable the final "collective decision" on the choice of the single-to-be-assigned label, given the outputs of the "dedicated per class" binary classifiers.

Specifically, our proposed architecture consists of a set of independent dedicated binary classifiers, each one with two output nodes, with each output node expressing the probability of the item under consideration to belong in the class. Our key intuition is that when only one of the independent dedicated classifiers, say k, puts a "large enough" probability on the item under consideration belonging to its class, while all others believe the item cannot be classified in their respective class, then we can be confident that the class to select as the output of our system is indeed the one predicted by the k-th classifier. This is because the classifiers' output essentially *mutually support* eachother. We act upon this intuition via a decision rule, *MSDR* (for "mutually supported decisions rule") we put forward, and which implements it.

Arguably, there are many settings and problems where the assumption behind this approach is valid and helpful; several such domains were mentioned in the beginning of this paper. Of course, there exist many cases in which features might be such that it is harder to distinguish among classes. That is, either the class boundaries are unclear, or the confidence of certain classifiers is low. The latter could occur for instance in the case of *highly unbalanced datasets*. Our approach is generic enough to tackle such cases, via the incorporation of *(i)* classifier-specific "confidence-related weights", and *(ii)* several alternative decision rules we propose, and which complement the main MSDR rule mentioned earlier.

Given the above, our main contribution is putting forward a generic architecture that explicitly exploits a "classifiers' mutual support" or "mutual exclusivity" property underlying many dataset domains of interest: the fact that in many cases an image in a given dataset might almost unquestionably belong to one class only. Our architecture is able to separate

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project code: T1EDK- 03110, ANASA).

individual features specific to each class, from the pool of shared features; and includes *designed-to-purpose decision rules* that are triggered when the mutual exclusivity property is or is not satisfied. To allow for flexibility and easy adaptation to the specific properties of the datasets of interest (e.g., on a dataset's degree of "balancedeness" with respect to the types of images it contains), our approach also makes use of "weights" associated to each independent classifier, and which intuitively mirror the confidence each dedicated classifier has in identifying its corresponding class. Our framework allows for the effective separation of items based on distinct class features; and is able, in its generality, to (a) take advantage of clear class boundaries when these exist, and (b) to effectively assign items to classes with increased confidence, even when clear class boundaries do not exist. Our experiments, conducted on a well-known dataset from the waste classification domain, confirm the effectiveness of our approach.

An additional, in a sense "emergent" contribution, is the fact that the proposed framework may be utilized to characterize a given dataset regarding its "homogeneity" and "balancedness" properties. That is, the framework can be used to apply a series of decision rules, potentially with varying weights and parameters and based on different intuitions and scenarios of interest. An exploratory qualitative and quantitative dataset analysis process can then be effectively carried out for each dataset, in parallel with and also following the classification decisions optimization one: by taking into account the frequency with which the conditions for the application of a specific *decision rule* is valid, one can analyze the dataset to decide its status—e.g., the degree in which contained items' classes are related to each other. This exploratory dataset analysis can be used to further drive the modeling process, for instance addressing certain questions regarding the choice of model parameters. For image classification problems, this may yield insights regarding the distributions of features, or for identifying meaningful trends of predictors in different classes. The same intuitions apply for generic classification problems: one can assess the use of simple "decision rules" to acquire meaningful knowledge regarding the dataset at hand, and further optimize the classification model.

II. Background and Related Work

As mentioned earlier, our architecture can be cast as an "one-vs-rest" classification scheme. Now, as is usually used in the literature, the term "one-vs-rest" describes a generic classification paradigm that uses binary classification algorithms for multi-class classification, and involves tackling the multi-class problem as multiple, *one-per-class* "binary" classification ones. A dedicated "binary" classifier that potentially uses a sigmoid function for classification, is then trained on the original dataset and solves a "binary" per-class classification problem; and final predictions are usually made using the classifier that is the most confident about the item being classified as a "yes" instance in its corresponding class, or by using a very simple voting rule [2], [3]. There are far too many examples of "one-vs-rest" classification algorithms to list here while doing them all justice. However, they all either output the prediction of the most confident classifier, or use a simple or absolute majority voting rule, as mentioned above. To the best of our knowledge, ours is the first approach in the classification literature that puts forward a generic framework encompassing multiple non-trivial decision making rules for the final classification. Several of our rules are inspired by approaches in *randomized social choice*, which employ rules that select winning candidates or outcomes from probability distributions over alternatives, in order to fend against strategic voting, promote participation in the choice mechanism, or to guarantee fairness in allocation problems [1].

III. System Architecture and Decision Rules

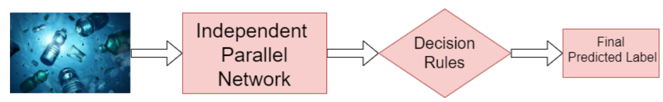

Fig. 1: Proposed Architecture

There is a set S_C of M classifiers, one dedicated for each class, following the one-vs-rest classification method for decomposing the multi-class problem to M dedicated binary classifiers. In departure to what is the norm in one-vs-rest classification approaches (in which the output of each binary classifier is usually a "yes/no" decision), the output of the network is a probability vector, each element of which contains the independent probabilistic estimate of each classifier representing its degree of certainty regarding an item belonging to its respective class. This vector will then be acted upon by the decision rules we put forward, and which we will be presenting in detail below, in order to come up with a final classification decision.

To explain further, Figure 2 depicts graphically the Independent Parallel Architecture in which we apply the different Decision Rules. The output of the architecture is a final vector of f_i^Y probabilities, signifying the degree of certainty by classifier i regarding an item belonging to its respective class; and our decision rules will be acting on this vector. We explain the f_i^Y probabilities, along with other required notation, immediately below:

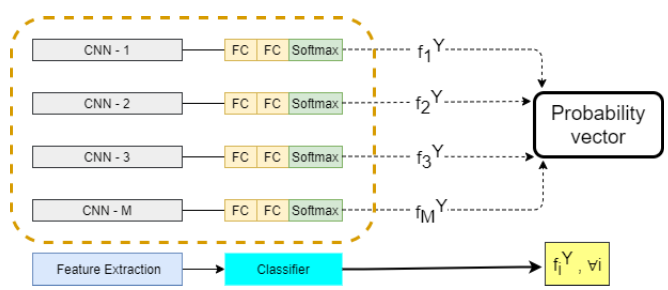

Fig. 2: Network of Independent Parallel CNN classifiers. FC = Fully Connected layer

Below we provide some required notation:

- $i \in M, |M| >= 2, M$: set of different classes

- p_i^Y : the dedicated classifier's output "yes" probability
- p_i^N : the dedicated classifier's output probability for "no"; $p_i^N = 1 - p_i^Y$
- w_i : the dedicated classifier's weight; $\sum_{i=1}^{M} w_i = 1$
- $f_i^Y = w_i \cdot p_i^Y$: class i's final "weighted probability" ; $f_i^N = 1 - f_i^Y$
- $\hat{f}_i^Y$: the normalized f_i^Y (probability) values of all classifiers ; $\hat{f}_i^N = 1 - \hat{f}_i^Y$
- $\tilde{f}_i^Y$: the normalized f_i^Y (probability) values of classifiers which output $f_i^Y > f_i^N$; $\tilde{f}_i^N = 1 - \tilde{f}_i^Y$
- $\bar{f}_i^Y$: the custom probability values (cf. Eq. 7) of classifiers which output $f_i^Y > f_i^N$
- c_k^s : the selected class according to a given rule k
- T : a required confidence threshold[1]

We note that deciding the w_i weights of each classifier is an interesting engineering problem: weights can represent our confidence to the specific classifier; or, more accurately, to the ability of the classifier to be effective on the particular dataset, given the features and other characteristics of the dataset—such as the number of its items, the degree of its items' homogeneity (e.g., features' distribution) and the number of items belonging to each possible class, and so on. As mentioned earlier, a dataset analysis process can be carried out multiple time for a particular dataset to optimize w_i's: once before classification to determine the original w_i weights, and then again in an iterative fashion, in order to re-determine the w_i following a given number of classifications (Fig. 3).

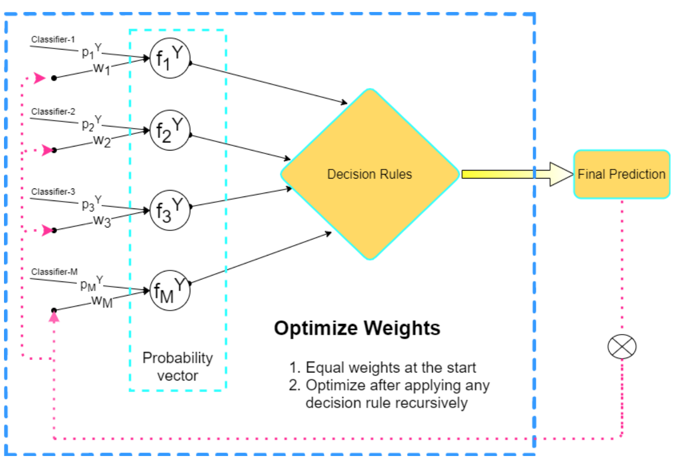

Fig. 3: Weight Optimization for the predictions after Applying Decision Rules

With this overall architecture and notation at hand, we now proceed to describe our opinion aggregation rules in detail.

A. Mutually Supported Decisions Rule (MSDR)

As mentioned earlier, our main decision rule is based on the key intuition that the findings of the various independent, dedicated classifiers are mutually supported, thus we term it *the Mutually Supported Decisions Rule (MSDR)*. We expect MSDR to be highly successful when only one classifier, k, predicted $f_k^Y > f_k^N$, and all others believe the item cannot be classified in their class, i.e., $f_i^Y < f_i^N, \forall i \neq k$. We can then be confident that the class c^s to select as our output is indeed the one predicted by the k-th classifier.

In 1 below we show our first of three MSDR variants. We represent the class selected by this decision rule by c_1^s, since it is the class returned by our first decision rule:

$$c_1^s = k, \quad \exists k : f_k^Y > f_k^N \;\&\; f_k^Y \geq T \;\&\; \forall j \neq k \; f_j^Y < f_j^N \quad (1)$$

That is, c_1^s is returned if one classifier only answers "yes" to the question on whether it believes the item belongs to the class of items it is trained to identify, and believes this with a confidence that exceeds a prespecified threshold. Therefore, we give this MSDR variant the name "MSDR-ONE-YES-THRESHOLD-YES" decision rule.

Now, we employ two alternative courses of action in case the main condition ($f_k^Y > f_k^N$ for just one k) holds in Eq. 1 above, but the threshold is not met. The first is to pick the class the k classifier predicts anyway. This is captured by Eq. 2, which thus constitutes our "MSDR-ONE-YES-THRESHOLD-NO-PICK-YES" decisions rule:

$$c_{1.1}^s = k, \exists k : f_k^Y > f_k^N \& \; f_k^Y < T \quad \& \; \forall j \neq k \; f_j^Y < f_j^N \quad (2)$$

The second alternative course of action for MSDR is to disallow the deterministic determination of the outcome by k. Instead, we normalize the f_i^Y values of all classifiers to respective $\hat{f}_i^Y$ values so as to sum to 1, and then randomly sample the class to be selected out of this distribution. This is our "MSDR-ONE-YES-THRESHOLD-NO-PICK-RANDOM" decision rule, shown below:

$$c_{1.2}^s = \arg random_{i \in S_C}(\hat{f}_i^Y), \quad \exists k : f_k^Y > f_k^N \& \; f_k^Y < T \;\&\; \forall j \neq k \; f_j^Y < f_j^N \quad (3)$$

Now, if the main MSDR condition of having only one dedicated classifier predicting its corresponding class *does not* hold, we use alternative rules and methods to pick a class to output as the one predicted by our system. These rules and methods are described in the following sections.

B. "Many yes" decision rules

When more than one classifiers have predicted a $f_i^Y > f_i^N$ probability, we can simply select the corresponding class with maximum probability, as long as a confidence threshold is met:

$$c_2^s = \underset{i}{\operatorname{argmax}} \, f_i^Y \quad \text{where } i : \quad f_i^Y > T \quad (4)$$

That is, we simply select the class c_2^s corresponding to the classifier with maximum f_i^Y across all classifiers, as long as $f_i^Y > T$. This is the "MANY-YES-THRESHOLD-YES-PICK-MAX" rule.

Now, if the threshold is not met, we are not confident enough to decide, so we can either *(a)* accept the class whose

[1]The default T value used in our experiments is 80%.

probability is highest, i.e., the one selected by the following "MANY-YES-THRESHOLD-NO-PICK-MAX" rule:

$$c^s_{2.1} = \underset{i}{\operatorname{argmax}} f^Y_i \tag{5}$$

or *(b)* we can normalize the f^Y_j values of only those classifiers with $f^Y_j > f^N_j$ to $\tilde{f}^Y_j$ so as to sum to 1, and then we randomly sample the selected class out of this distribution. This is the "MANY-YES-THRESHOLD-NO-PICK-RANDOM":

$$c^s_{2.2} = \arg random_j(\tilde{f}^Y_j) \tag{6}$$

As mentioned, the $\tilde{f}^Y_j$ vector contains the normalized softmax elements of the classifiers' final output probability $\forall j \in C_S$ with $f^Y_j > f^N_j$. So, $c^s_{2.2}$ is randomly chosen from this particular subset of classes, according to their $\tilde{f}^Y_j$ probabilities.

C. Weighted Soft Voting

When more than one classifiers have predicted a $f^Y_i > f^N_i$ probability, we implement a *weighted soft voting* method utilizing class-specific weights w_i, and calculate new custom probabilities only for these classifiers. We predict the custom probabilities $\bar{f}^Y_i$ only for the classifiers which outputs $f^Y_j > f^N_j$ with the help of the others classifiers which output $f^Y_j < f^N_j$. That is, for every classifier which outputs $f^Y_i > f^N_i$ the custom probability $\bar{f}^Y_i$ is calculated according to the equation below:

$$\bar{f}^Y_i = w_i \cdot p^Y_i + \sum_{j:f^Y_j < f^N_j} w_j \cdot p^N_j, \quad \forall i : f^Y_i > f^N_i \tag{7}$$

and then we pick the highest among them; this is the "MANY-YES-WEIGHTED-SOFT-VOTING" decisions rule:

$$c^s_3 = \underset{i:f^Y_i > f^N_i}{\operatorname{argmax}} \bar{f}^Y_i \tag{8}$$

As mentioned, the $\bar{f}^Y_j$ vector contains the custom probabilities elements of the classifiers' final output probability, $\forall j \in M$ with $f^Y_j > f^N_j$. So, c^s_3 is selected from this particular subset of classes, according to their $\bar{f}^Y_j$ probabilities.

D. Negative Predictors Rule

Now, when all classifiers predict $f^Y_i < f^N_i$—i.e., when they are all "pessimistic" or "negative" regarding the item under consideration falling in their corresponding class—we simply choose the class with highest f^Y_i probability :

$$c^s_4 = \underset{i:f^Y_i < f^N_i}{\operatorname{argmax}} f^Y_i \quad \text{, used when all } i \text{ predict } f^Y_i < f^N_i \tag{9}$$

This is the "ALL-NO-PICK-MAX" decision rule.

E. Decision Rules Application Examples

The decision rules discussed in this chapter are summarized in Table I. These decision rules can help resolve indecisive cases such as the ones in Tables II and III.

TABLE I: Decision Rules Description Summary

Decision Rule	Description	Eq.
MSDR-ONE-YES-THRESHOLD-YES	Only one classifier considers the item to belong to its class with enough confidence	1
MSDR-ONE-YES-THRESHOLD-NO-PICK-YES	Only one classifier considers the item to belong to its class without enough confidence; pick that class	2
MSDR-ONE-YES-THRESHOLD-NO-PICK-RANDOM	Only one classifier considers the item to belong to its class without enough confidence; pick randomly among all classifier outcomes after normalization	3
MANY-YES-THRESHOLD-YES-PICK-MAX	More than one classifiers consider the item to belong in their class, and do so with enough confidence; pick the one with highest confidence	4
MANY-YES-THRESHOLD-NO-PICK-MAX	More than one classifiers consider the item to belong in their class, but do so without enough confidence; pick the one with highest confidence	5
MANY-YES-THRESHOLD-NO-PICK-RANDOM	More than one classifiers consider the item to belong in their class, but do so without enough confidence; select randomly among them, after normalization	6
MANY-YES-WEIGHTED-SOFT-VOTING	More than one classifiers consider the item to belong in their class; select the one among them with the highest custom probability given by Eq. 7	8
ALL-NO-PICK-MAX	All classifier consider the item to not belong in their class and we simply pick the smaller score	9

TABLE II: Applying different MSDR-ONE-YES rules

Classifier: f^Y_i	Probability, T=90%		
	Case 1	Case 2	Case 3
1	$\boxed{0.9}$	0.3	0.23
2	0.4	$\boxed{0.84}$	0.3
3	0.3	0.4	0.8
4	0.35	0.3	0.4
5	0.2	0.35	0.3

In Table II we see three cases in which we apply a different "MSDR" decision rule. In more detail, given the f^Y_i for every classifier, $i \in [1, 5]$, we select the appropriate rule. For example in *case 1* we select the class of dedicated classifier "1" following Eq. 1, since its confidence surpasses the $T = 90\%$ threshold. In *case 2* the confidence of the one classifier that predicted $f^Y_i > f^N_i$, here "2", does not surpass T, but we choose to pick it simply by following Eq.2. Finally in *case 3*, where again one classifier believes the item belongs to its class, but with a lower than T probability, we normalize all probabilities and sample following Eq. 3. *Cases 2 and 3* both fall under the "MSDR-one-yes-theshold-no" category, but we have applied a different rule (*MSDR-one-threshold-no-pick-yes* vs *MSDR-one-threshold-no-pick-random*) in each case.

TABLE III: Applying different MANY-YES rules

Classifier: f^Y_i	Probability, T=90%		
	Case 4	Case 5	Case 6
1	0.91	0.3	$\boxed{0.63}$
2	0.4	$\boxed{0.84}$	0.3
3	$\boxed{0.92}$	0.4	$\boxed{0.86}$
4	0.35	0.67	0.4
5	0.2	0.35	$\boxed{0.78}$

Table III lists three cases in which we apply a different "MANY-YES" decision rule. In more detail, given the f_i^Y for every classifier, $i \in [1, 5]$, and a 90% confidence threshold, we select a classifier given the rule appropriate in each case. For example in *case 4* we select the classifier-3 predicted class following Eq. 4 due to the fact that the classifier has the highest probability among the classifier which predicted $f_i^Y > f_i^N$ and have met a 90% threshold. In *case 5* we follow Eq. 5 and select according to the classifier with the highest confidence, here classifier-2. Finally, in *case 6*, we follow the approach of Eq. 6): we normalize all probabilities from classifiers which predicted $f_i^Y > f_i^N$, and use sampling to assign one of the corresponding classes to the item.

TABLE IV: Performance of the individual Independent Parallel classifiers and the Baseline CNN (during training)

Model Name	4-fold Accuracy	Specificity	Sensitivity
Baseline_CNN	89.58% (+/- 1.24%)		
Cardboard classifier	96.94% (+/- 0.88%)	99%	78%
Glass classifier	93.40% (+/- 0.64%)	97%	78%
Metal classifier	93.30% (+/- 1.04%)	97%	84%
Paper classifier	96.42% (+/- 0.48%)	98%	87%
Plastic classifier	93.68% (+/- 1.13%)	98%	79%

IV. Experiments

We evaluate our framework by running tests on a well-known dataset from the waste classification domain. We chose to focus on a slightly *unbalanced* dataset. A dataset is considered "balanced" when it contains an (approximately) equal number of images in each class. By contrast, *unbalanced* datasets exhibit singificant variance in the number of examples in each class of the problem [4]. Unbalanced datasets pose a challenge to predictive modeling, since most of techniques assume an equal number of instances for each class; typically, models trained on unbalanced datasets have poor predictive efficiency, especially for the minority class [5]–[7].

Our dataset of choice is the *TrashNet* data collection [8]. It consists of images of six classes: Glass, Paper, Cardboard, Plastic, Metal and Trash. We focus our attention on images labeled as contained in one of the five first classes (i.e., we exclude Trash). The final dataset consists of 799 images: 140 cardboard, 165 glass, 142 metal, 190 paper, and 162 plastic; and is thus a "slightly" unbalanced dataset.

For our experiments, we first train a *baseline CNN* model over a "slightly unbalanced" training set of 530 TrashNet images; and also the *binary* Independent Parallel (IP) CNN classifiers over their respective sub-datasets making up the training set (specifically, the training set contains 87 cardboard, 112 glass, 89 metal, 135 paper, and 107 plastic images). The *baseline CNN* and the IP CNNs all follow the ResNet50 [9] architecture (but with the baseline CNN performing multiclass classification, i.e., having as many output units as the number of classes). The accuracy the baseline model and that of each binary IP classifier, along with other metrics for the IP-CNNs during training are shown in Table IV; while Table V shows the performance of the *baseline CNN* on each class. We can see that the overall accuracy of the *baseline CNN* during training on TrashNet is 3% lower than the lowest accuracy exhibited by the IP classifiers.

TABLE V: Baseline CNN performance on each class (training)

Class Name	Specificity	Sensitivity
Cardboard class	98%	93%
Glass class	92%	91%
Metal class	96%	85%
Paper class	98%	91%
Plastic class	98%	78%

We then use the IP CNNs above as "frozen classifiers", add the decision rules application layer to the netwrok of the IP CNNs, and evaluate the performance of our overall decisions rules-based framework against that of the baseline CNN in a hold-out set of unseen images.

To begin, in Tables VI and VII we can see the classification report metrics over a TrashNet testing set of 269 images, for *the baseline network* and *our complete decision rules-based framework*[2] respectively. The results show that almost in every metric we observe a increase of 1-4 % wrt the baseline model. Notably, we have a 4% increase in overall accuracy.

TABLE VI: Classification Report for Baseline CNN (testing)

Material	Precision	Recall	f1-score	support	Specificity	Sensitivity
cardboard	94%	92%	93%	53	98.6%	92.4%
glass	75%	89%	81%	53	92.6%	88.6%
metal	82%	85%	83%	53	95.3%	84.9%
paper	74%	91%	81%	55	91.6%	90.9%
plastic	77%	44%	56%	55	96.8%	43.6%
accuracy			80%	269		
macro avg	80%	80%	79%	269		
weighted avg	80%	80%	79%	269		

TABLE VII: Classification Report for the overall Decision Rules-based Framework (testing phase)

Material	Precision	Recall	f1-score	support	Specificity	Sensitivity
cardboard	94%	94%	94%	53	98.6%	94.3%
glass	82%	87%	84%	53	95.3%	86.7%
metal	83%	92%	88%	53	95.3%	92.4%
paper	75%	91%	82%	55	92.2%	90.9%
plastic	88%	55%	67%	55	98.1%	54.5%
accuracy			84%	269		
macro avg	84%	84%	83%	269		
weighted avg	84%	84%	83%	269		

Finally, the confusion matrices in Figures 4 and 5 demonstrate the ability of our framework to "correct" the failure of the baseline CNN to correctly identify several plastic items.

[2]In our reported experiments we used rules corresponding to Equations 1, 2, 4, 5 and 9 (rules' "Combination 1"). Results obtained so far using different combinations of rules are similar to the ones reported, but we do not report these here due to space restictions.

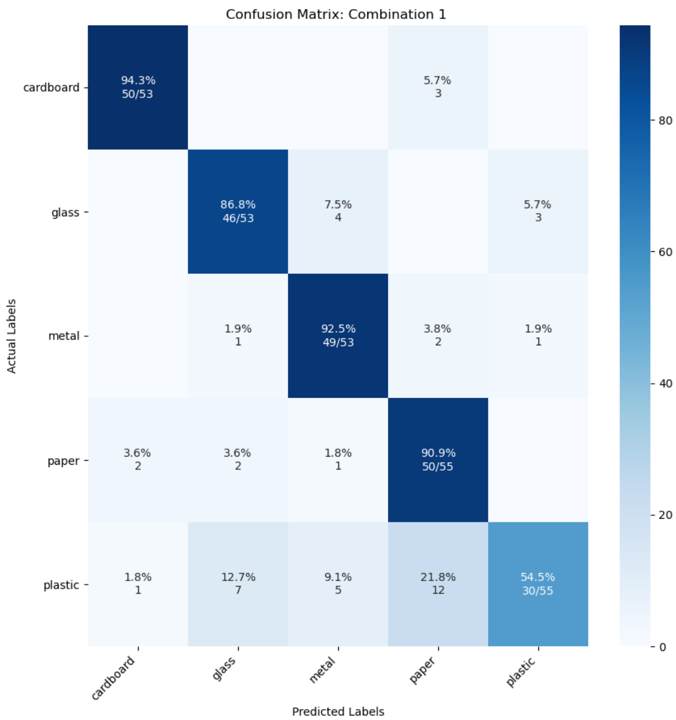

Fig. 4: Decision Rules Framework Conf. Matrix (testing)

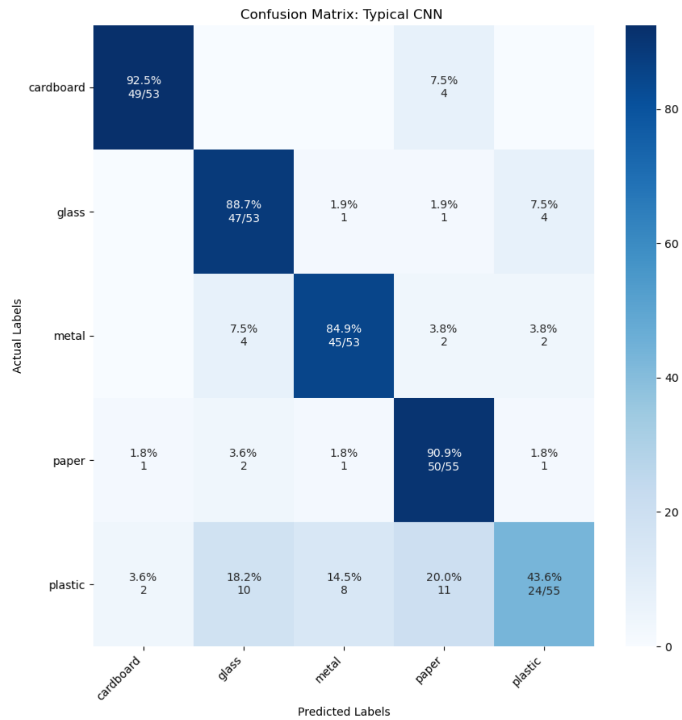

Fig. 5: Baseline CNN Conf. Matrix (testing)

V. Conclusions and Future Work

In this work we put forward a novel, generic one-vs-rest classification framework that encompasses multiple non-trivial decision rules to effectively aggregate the opinions of several independent parallel CNNs. Some of the rules proposed, explicitly exploit a mutual exclusivity property underlying many image datasets of interest; while others are inspired by randomized social choice schemes and assign a class to the item under consideration via picking from a distribution over alternatives. The framework takes advantage of clear class boundaries when these exist, while being able to assign items to classes with increased confidence, even when clear class boundaries do not exist. Our experiments with the well-known TrashNet dataset, confirm the potential of the approach.

The proposed framework opens up research avenues for much interesting future work. To begin, the effectiveness of various combinations of decision rules (e.g., combining max-picking rules with randomized ones in different cases) has to be verified via systematic experimentation on different datasets of interest; and statistics on the frequency with which different rules are triggered in different datasets need to be accumulated. Results obtained in this manner can lead to the implicit characterization of a dataset, in terms of "balancedness", "homogeneity", or "easiness" (e.g., if an "MSDR-one-yes" rule frequently applies, this can be a signal that a dataset largely possesses the mutual exclusivity property). Finally, coming up with effective and efficient ways to set and re-adjust the weights of the independent parallel classifiers poses an interesting problem; to this end, we intend to exploit uncertainty metrics originating in the *scoring rules* literature [10], to objectively characterize our confidence on a classifier's performance.

References

[1] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Eds., *Handbook of Computational Social Choice*. Cambridge University Press, 2016. [Online]. Available: https://doi.org/10.1017/CBO9781107446984

[2] A. C. Lorena, A. C. P. de Leon Ferreira de Carvalho, and J. Gama, "A review on the combination of binary classifiers in multiclass problems," *Artificial Intelligence Review*, vol. 30, no. 1-4, pp. 19–37, 2008. [Online]. Available: https://doi.org/10.1007/s10462-009-9114-9

[3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[4] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 11.

[5] P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modelling under imbalanced distributions," 2015.

[6] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, pp. 221–232, 2016.

[7] H. He and Y. Ma, *Imbalanced learning: foundations, algorithms, and applications*. Wiley - IEEE Press, 2013.

[8] M. Yang and G. Thung, "Classification of trash for recyclability status," in *Sanford CS229 Project Report*, 2016. [Online]. Available: https://github.com/garythung/trashnet

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA*, 2016, pp. 770–778.

[10] T. Gneiting and A. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, pp. 359 – 378, 2007.