Extracting Features with Deep Learning for Ensemble-Driven Case-Based Classification

Zachary Wilkerson[1(✉)], David Leake[1], David Crandall[1], and Benjamin Wilkerson[2]

[1] Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington 47408, IN, USA
zachwilk@iu.edu

[2] DePauw University, Greencastle 46135, IN, USA

Abstract. Manual knowledge acquisition of case retrieval features is expensive and may be infeasible for cases containing hard-to-characterize data such as images. Deep learning (DL) methods excel at extracting useful feature information from raw data, making them appealing for learning feature information. Previous work has demonstrated the promise of integrated systems for case-based image classification, using a deep neural network to generate features which are then used for case retrieval, resulting in classifications that can be explained in terms of prior cases. However, the accuracy of the combined system may lag behind that of the original DL model. In response, our previous work proposed Multi-Net, a method using ensembles for localized feature extraction. Multi-Net improved performance, but experiments showed limitations of its design. This paper presents Deep Ensemble Feature Extraction for Retrieval (DEFER), a feature-extraction-based classification approach aimed at addressing those issues. To increase accuracy, DEFER adds a discriminator to focus retrieval within each replica and weighted voting based on confidence in its class prediction, grounded in nearest-neighbor retrieval. In experiments for image classification, DEFER outperforms analogous DL-only and DL-case-based systems, supporting that its approach can improve performance.

Keywords: Case-Based Reasoning · Deep Learning · Ensembles · Feature Learning · Indexing · Integrated Systems · Random Forests · Retrieval

1 Introduction

Case-based classification depends on effective case retrieval, which in turn typically depends on the feature representation of cases in the case base. Traditionally, such features are drawn from a vocabulary defined by knowledge engineering (e.g., [14,20,31]). However, such an approach is costly and may not

I. Bichindaritz and B. López (Eds.): ICCBR 2025, LNAI 15662, pp. 345–359, 2025.
https://doi.org/10.1007/978-3-031-96559-3_23

produce comprehensive feature vocabularies for poorly understood domains or for tasks for which effective features are hard to characterize, such as image classification. This motivates efforts to apply machine learning to deriving case features and assessing case similarity (e.g., [30,33,35]). Alternatively, in twin systems, case retrieval features are extracted from a network to explain network outputs with cases [19]. Our previous work treats case-based classification as the primary classification mechanism, providing interpretability, while leveraging features extracted from deep learning (DL) neural networks to improve case retrieval [21,22,34,35]. This integrated model has been shown to outperform an analogous DL-only model in previous studies for image classification (e.g., [35]).

Our initial integrated methods were based on training a network and then extracting features from a network layer. While this approach provided good performance, it had two primary limitations. First, extracting large numbers of features commonly generated by deep neural network models may result in similarity judgments suffering from the "curse of dimensionality" [21]. Second, we hypothesized that performance of the case-based classifier might be limited by a mismatch between the needs for DL features—which drove network training—and the needs of case-based reasoning (CBR) features. Intuitively, neural models rely on identifying features suited to distinguishing different classes (i.e., strong inter-class discrimination); by contrast, CBR models require features suited to recognizing similarity between examples that belong to the same class (i.e., strong intra-class relationship identification). To the extent that this holds, features extracted from neural models trained for classification might not be optimal for use in case retrieval.

To address the two limitations, we proposed the "Multi-Net" architecture for localized feature extraction. Multi-Net breaks up the overall classification task into smaller class-wise tasks [21]. In Multi-Net, multiple replica neural models are trained to distinguish examples belonging to one class from all other classes. Ideally, this requires less training data, enables lower-dimensionality network construction for feature extraction, and exploits localized representations for greater classification accuracy. These benefits appeared to hold for initial work. However, preliminary tests with more complex, pretrained DL models showed that Multi-Net underperformed relative to our other DL-CBR approaches and that replica training in Multi-Net resulted in independent feature spaces with non-corresponding features. This made nearest-neighbor case retrieval, which attempted to leverage these features collectively, ineffective.

This led us to develop a new model, presented in this paper, that replaces the Multi-Net approach with an ensemble-based method inspired by random forest approaches. In this approach, the goal for the replica models is to learn projections of the overall feature space that inform per-replica nearest-neighbor case retrieval; each result then contributes to a majority vote for the overall classification. The model also dynamically selects subsets of the case base to consider and uses measurements of the confidence of retrieved nearest neighbor cases to further guide case-based classification. As this process defers CBR until after dispatching to the replicas, we call the approach Deep Ensemble Feature Extraction

for Retrieval (DEFER). This paper presents a case study evaluation of DEFER. In this study, testing on the Animals with Attributes 2 (AwA2) dataset [36], it consistently outperforms the top model from our previous work [35] and shows stronger classification accuracy than the analogous DL-only model, with Student's T-test statistics supporting the significance of this improvement.

2 Related Work

2.1 DL-CBR Integrations

CBR and DL approaches present complementary strengths that make their integration appealing. When integrations base their reasoning fundamentally on CBR but leverage DL, such combinations can provide more explainable alternatives to DL-only models and increased ability to integrate expert knowledge into various knowledge containers [29]; ideally the use of DL to support CBR may enable high accuracy and minimal knowledge engineering in big data domains. Some integration approaches aim to model a CBR process within a network, as in "inherently interpretable networks" [5,10,24] that make predictions based on similarity scores between intermediate features within the network and sets of prototype features that align well with certain classes (e.g., a beak for classifying an image of a bird), and in the NN-kNN model [38]. Case-based approaches may also be applied post-hoc to provide explanations for DL models; this has been studied for feature-level explanations [3] and applying DL and CBR models in parallel "twin systems" to explain DL predictions [19]. Conversely, network models have been used within CBR for tasks such as similarity assessment (e.g., [26]) and case adaptation (e.g., [23]).

2.2 Feature Extraction for Case Retrieval

Traditionally, retrieval features are created via knowledge engineering [14,20,31], with symbolic learning methods sometimes applied to develop or refine feature vocabularies for domains where knowledge engineering is unfeasible [4,6,7,11,15]. However, such approaches apply only to domains where symbolic knowledge is available; for others, such as image classification, it is necessary to extract features from raw data in order to develop the feature vocabulary. In response, multiple projects have explored extracting retrieval features from DL models (e.g., [2,30,33]). Features extracted from DL models in this way have been used for CBR-based implicit classification of novel-class test examples and examples for which the DL model lacks confidence in its own classification [32,33]. At least for some domain examples, CBR classification using extracted features has outperformed DL-only predictions [30,35].

Our previous work studied feature extraction from DL networks, especially for small-data domains, to which CBR systems are frequently applied, studying the influence of the layer chosen for extraction and the number of features extracted on DL model convergence and CBR classifier performance [21]. It also considered the impact of different model-level parameterizations, such as the

choice of DL architecture and the use of pretraining, highlighting their importance with respect to extracted feature quality [21,22,35], as well as how engineered features may be used in concert with extracted features to increase model accuracy [22,34]. This paper refines and extends the Multi-Net approach [21], described in detail in Sect. 3.1.

2.3 Random Forests and Ensemble-Based CBR Approaches

An issue for our initial Multi-Net model was that each network developed an independent feature space, so that when comparing distances to candidate neighbor cases during KNN, the localized feature spaces had different numeric scales. We hypothesized that the lack of correspondence between replica feature spaces decreased the usefulness of combined distance information, which motivates our shift from the locality-driven Multi-Net model to the ensemble-driven case retrieval of DEFER. The DEFER model draws significantly from random forest principles [8], in that the replica models can be conceptualized as learning projections of an overall feature space, analogous to the randomized projections leveraged in a random forest. Similar principles have previously been applied to cases by "Random KNN" approaches, which use the random forest conceptualization for feature reduction in the KNN feature space [25].

Ensemble-based retrieval also relates to the work of Plaza and others on distributed and multi-agent CBR [27,28]. In that work, the CBR algorithm draws from the decisions of multiple agents, which each may consider subsets of the CBR cycle and/or exploit cases from multiple non-corresponding case bases. Similarly, other combinations of CBR and ensemble techniques are present in work by Hsieh et al., where CBR is leveraged to mediate disagreements within the ensemble [17]. Again, a key interpretation of DEFER's feature extraction process is that it learns different projections of the feature space that are well-suited to identifying subsets of classes. This aligns with the work of Cunningham and Zenobi, who focus on diversity of features developed for CBR ensembles to promote specialization in "sub-domains" within the feature space [12].

3 Feature Extraction for CBR Retrieval using Multiple Networks

3.1 A Multi-network Approach

We developed the Multi-Net approach to address two issues: 1) that DL models require a significant number of parameters to model multi-class functions, and this high dimensionality is passed on to the extracted feature vectors, risking the curse of dimensionality for CBR systems that use them, and 2) that neural models ideally learn generalizable features that help them discriminate between all classes in their search space, but we hypothesized that this high degree of inter-class learning may come at the expense of intra-class learning, desirable for

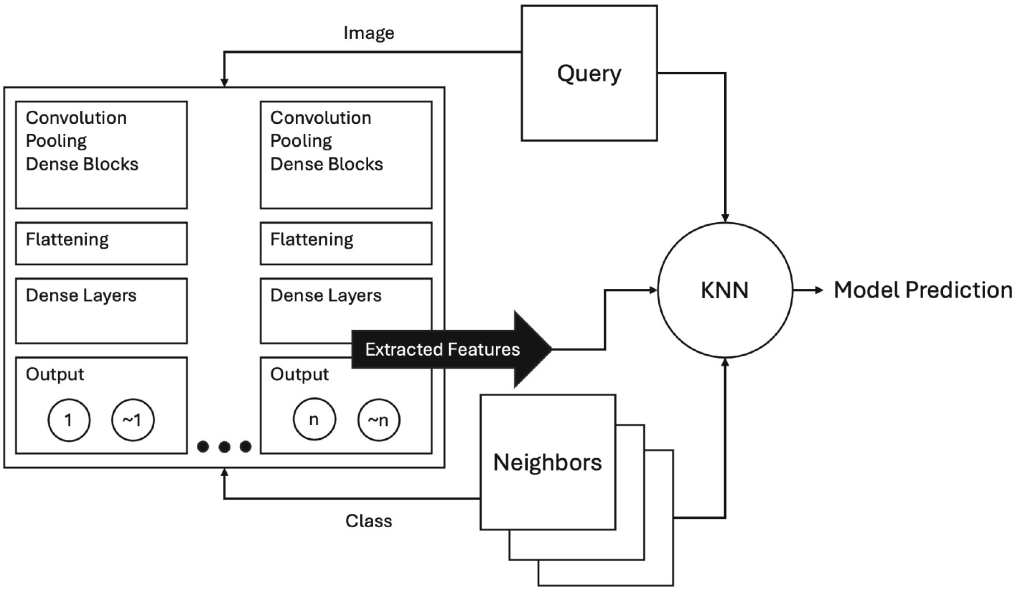

Fig. 1. Data flow in the original Multi-Net methodology [21]. Each query image is processed into localized features by each replica; this set of features is indexed during KNN based on the candidate neighbor's class for similarity calculation.

CBR. This motivated an architecture splitting the original problem into several smaller problems involving fewer classes, using specialized neural model replicas to address these smaller problems [21].

Rather than training one neural model that can distinguish between n classes, Multi-Net trains n neural models that each distinguish between a single positively-labeled class and all others in the dataset, which are relabeled as negative classes (Fig. 1). In this way, replicas are specialized to the positive examples in their training set, at the additional overhead of training n neural models rather than one. Thus, n different feature vectors may be extracted for a single test example, one for each replica. During KNN retrieval, each candidate neighbor's ground truth class indexes the space of query feature vectors, ideally creating a localized feature representation during similarity assessment that calculates distances to a candidate case as if the query belongs to the same class.

Strengths and Limitations: Multi-Net requires fewer features per replica to accurately characterize the dataset, but at the cost of additional space and training time proportional to the number of replicas used. In addition, relabeling the positive and negative training examples for each replica creates a class imbalance with many more negative classes, limiting accuracy. We address this in DEFER by allowing replicas to be sensitized to subsets of positive classes rather than only one, increasing the ratio of positive to negative examples for each replica while still retaining benefits of using multiple neural models.

An important additional concern for Multi-Net is that the replica models each learn independently from one another, so there is no correspondence between

their feature spaces. Consequently, individual feature values can be significantly larger or smaller for one given replica than for all others, making it impossible to reliably compare distances to nearest neighbors across all replicas—a candidate neighbor handled by a replica that learns a small-scale feature space might erroneously be labeled as the nearest neighbor simply because of its replica's smaller scale. Normalizing the feature spaces for the neural models does not solve this issue, as there is no correspondence between individual features between different replicas. This lack of correspondence was not evident in earlier results [21], but we hypothesized it as the cause when preliminary experimental results for the work in this paper showed that feature extraction from more complex pretrained neural models had a low performance ceiling compared to the analogous integrated system that we use as a baseline DL-CBR model [35]. We developed the DEFER approach, summarized in Fig. 2 and Algorithm 1, to address these shortcomings.

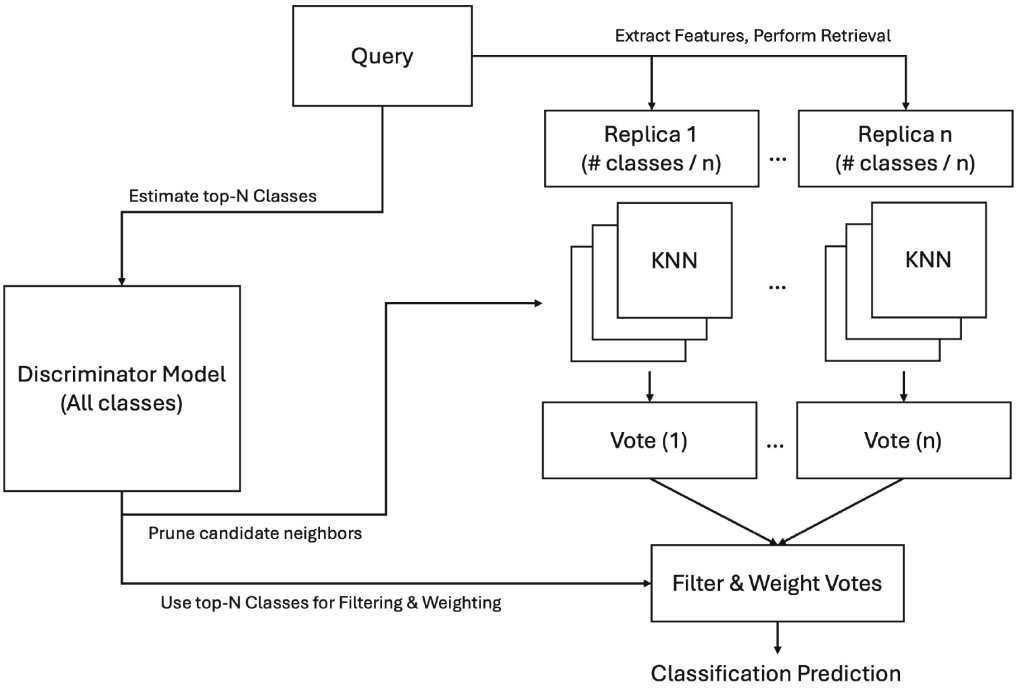

Fig. 2. Data flow in the DEFER model. Each query image is passed into each replica to generate localized feature vectors and a discriminator model trained on all classes. Each replica may only retrieve nearest neighbors belonging to the top-n discriminator predictions. Replica votes are weighted according to the ratio of distances to the two nearest different-class neighbors.

3.2 Leveraging the Spirit of Random Forests

In the DEFER approach, rather than treating the replicas as means for generating localized features for use in a global similarity assessment, the replicas

```
1: cpr ← number of classes per replica
2: cb ← input case base
3: D ← discriminator model
4: for r in 0 to number of replicas −1 do
5:     for (example, label) in training data do
6:         if floor(label / cpr) == r then
7:             label ← label mod cpr
8:         end if
9:     end for
10:    replica[r] trained with modified (examples, labels)
11: end for
12: D trained with original (examples, labels)
13: for (example, label) in training data do
14:    f ← features extracted from replica[label % cpr]
15:    add case(f, label) to cb
16: end for
17: for query, label in testing data do
18:    votes ← []
19:    C ← top n predicted classes from D(query)
20:    for r in 0 to number of replicas −1 do
21:        f ← features extracted from replica[r]
22:        q ← case(f, _)
23:        nn_1 ← nearest retrieved neighbor case, where nn_1 ∈ C
24:        nn_2 ← nearest retrieved neighbor case, where label_{nn_1} ≠ label_{nn_2}, nn_2 ∈ C
25:        if ∃c ∈ C where floor(c / cpr) = r then
26:            append nn_1 to votes with weight (distance to nn_2) / (distance to nn_1)
27:        end if
28:    end for
29:    prediction ← majority vote among votes
30: end for
```

Algorithm 1: DEFER algorithm for data relabeling and replica training (lines 4-11), discriminator-guided KNN for each replica (lines 19-28), and ratio-weighting (lines 23-26), all supporting majority-vote classification.

are treated as members of a random forest-like ensemble of elements performing case-based classification using their own feature spaces. The independent feature spaces learned by the individual replicas can be thought of as subsets of a single global feature space, similar to the randomly selected subsets represented in a random forest. Each replica then performs KNN and contributes a vote to the ultimate model decision (Algorithm 1, lines 17–30).

Strengths and Limitations: Performing replica-wise KNN calculations that result in individual votes addresses the issue of the replicas learning independent feature spaces with different scales, and it enables explaining classifications from the perspective of each of the different feature spaces. However, we empirically observed in preliminary experiments that the majority-vote system can suffer from significant replica-based noise in practice. That is, for a given query, only one replica is sensitized to the corresponding class; for all others, the class was relabeled as negative during training, and so there is no incentive for the replica to distinguish between it and any other negative class. In theory, this means that the one replica might provide useful information with its vote, but the other replicas could create a "tyranny of the majority" with contradicting, low-

confidence votes. This phenomenon requires additional refinements to achieve accurate classification performance, which are described below.

3.3 Focusing on the Top-n Cases with Discriminator Pruning

In addition to the random forest conceptualization for handling the different replicas, we implement a "discriminator model" to reduce disagreement among replicas by pruning the search space of candidate neighbors and funneling replica votes into a smaller field of classes. Intuitively, for a given query case, much of the case base is not relevant, so applying the discriminator enables DEFER replicas to focus on subsets of the case base that are likely more relevant to classifying a given query. This discriminator is a regular DL model with the same basic architecture as the replicas, except that it is trained to predict any of the classes in the dataset. Critically, instead of taking a single query classification from the discriminator, DEFER takes the top n classifications. These inform pruning during the per-replica KNN procedures at two levels (Fig. 2 and Algorithm 1, lines 13, 19–27):

1. Replicas only consider stored cases if their class is one of the top n classes provided. This streamlines the KNN calculations for each replica and ideally results in more agreement among the replicas, which have a less-diverse field of potential classes to choose from when voting.
2. When tallying votes, only votes from replicas that are sensitized to a subset of these top n classes are considered. This helps mitigate noise that results from low-confidence votes from other replicas.

Strengths and Limitations: Use of the discriminator incurs additional training overhead but significantly speeds up the model evaluation process because of pruning. Ideally, it also improves model accuracy over the similarly-structured baseline DL-CBR model by enabling the consideration of the top n predictions as opposed to one, enabling more subtle conclusions as secondary choices contribute. However, if the ground truth class is not in the discriminator's top n predictions, then the overall model cannot make a correct prediction, creating a hard accuracy ceiling for DEFER.

3.4 Favoring High-Confidence Cases with Ratio-Weighted Voting

Preliminary experiments with different values for n for taking the top n discriminator predictions suggested that $n = 2$ was most effective. However, during replica voting, sometimes either one replica is sensitive to both classes, or two are sensitive to one each. The latter scenario could lead to a tie that must be broken if the two replicas disagree.

To address this, we weight each contributing replica's vote based on an estimate for how confident the replica is that the query belongs to the neighbor's class, calculated using a ratio of the distances to the nearest neighbors of the two most relevant classes to the query (Fig. 2 and Algorithm 1, lines 23–26). A large

ratio signifies higher confidence, as the nearest neighbor is significantly closer to the query than any case of another class; conversely, a ratio approaching 1 implies low confidence, since there is more significant overlap between the class clusters in the feature space.

Strengths and Limitations: This approach provides a confidence score for a replica's decision, usable to break ties for low values of n, and it also offers a degree of explanatory power by providing some insight into the clustering of cases belonging to a single class (or lack thereof) in the feature space of a given replica. In preliminary tests, replicas not sensitive to any of the top n discriminator predictions frequently had ratios close to 1; this motivated our design decision to remove their votes. In principle, the reliance on the two closest cases might make classification more sensitive to noise if noisy cases are present, but we expect increased accuracy otherwise.

4 Evaluation Design

4.1 Motivation and Hypotheses

While the design of DEFER was informed by a series of preliminary experiments, we evaluate the final model by testing the following hypotheses:

1. **DEFER will outperform an analogous regular DL-CBR architecture.** By leveraging the benefits of ensembles to solve problems, DEFER should outperform both the analogous DL-only model and the analogous baseline DL-CBR integrated model.
2. **Each of the design choices for DEFER contributes to its superior classification performance versus the evaluated baseline models.** These choices include the ability to select multiple replicas (and by extension, the number of classes to which each one is sensitized), the use of the discriminator and ability to set the value of n for the top n classes taken from it, and ratio-weighting for replica votes.

4.2 Testbed System and Evaluation Method

Our tests use the pretrained DenseNet121 [18] neural architecture provided in the Tensorflow Applications module [1] for the DL feature extractor and for the discriminator. The CBR classifier component of the model is retrieval-only and uses 1-nearest-neighbor retrieval with an unweighted and normalized Manhattan Distance as distance measure. We note that a classification adaptation component (e.g., [37]) could be added with no changes to either of the case-based processes, potentially increasing accuracy of each.

We compare DEFER to two baselines: a DL-only model (i.e., in which the model's prediction is used directly, with no feature extraction or case-based component), and the single-network DL-CBR model from Wilkerson et al. [35]. All neural models are the same except for the size of the output layer, which depends on the number of classes used to train the network, and the number of training

examples is varied as a control and variable of comparison for each method. Features are extracted as vectors of size 1024 following the densely-connected layers of the neural model in all experiments using case-based classification.

Each neural model and replica is pretrained on the ImageNet dataset [13] as provided in the Tensorflow applications module. Post-convolution layers are appended and trained manually for every model and replica, while all convolution layers are frozen during training. Training and evaluation are carried out using the Animals with Attributes 2 (AwA2) dataset, which contains over 37,000 images depicting animals that are categorized into one of fifty classes [36]. All models or replicas are trained for a maximum of 50 epochs, with training halted if accuracy does not improve for two consecutive training epochs. Trials are run for 512, 1024, 2048, and 4096 training examples. Train and test examples are selected randomly and independently from the larger dataset, with test examples used for validation during training. Furthermore, positively-relabeled training examples are oversampled at a 200% rate to mitigate harmful data imbalance effects. Oversampled data are also selected randomly as part of the training set, but alternative techniques such as SMOTE could also be used [9]. For each MultiNet replica, the training set is relabeled such that a subset of all classes retains their original labels and all other labels are treated as belonging to another negative class. These subsets of classes are selected in the same arbitrary order from the training directory for consistency among trials, and such that classes are distributed evenly among the replicas. All trials are repeated thirty times to establish reliable mean and standard deviation values.

5 Results and Discussion

5.1 Tests of Hypothesis 1: Comparison to Baseline Models

As shown in Fig. 3, DEFER outperforms our baseline DL-CBR model on average for all numbers of training examples. This result is noteworthy because the latter model was previously found to outperform the analogous DL-only model with similar consistency across all training examples in the same case study [35]. However, the previous outperformance was often subsumed by the standard deviation values for both models, limiting significance; DEFER's performance is more accurate and includes some results are significantly higher than the DL-only model's performance; this strongly supports our first hypothesis.

Comparing DEFER and our 2024 model, the average standard deviation of ensemble-driven case retrieval appears to be generally higher; this is not surprising, given that each replica during training receives an imbalanced dataset, and therefore, fewer "useful" training examples. The mean accuracy of DEFER consistently exceeds the DL-CBR baseline. This difference appears to decrease slightly as the number of training examples increases; however, Student's T-test applied to DEFER versus either of the other two models provides p-scores of $p \leq 0.0035$ across all data points, suggesting a high confidence in the significance of DEFER's performance, compared with the typical $p \leq 0.05$ threshold. Taken together, these results broadly support and expand upon the conclusions

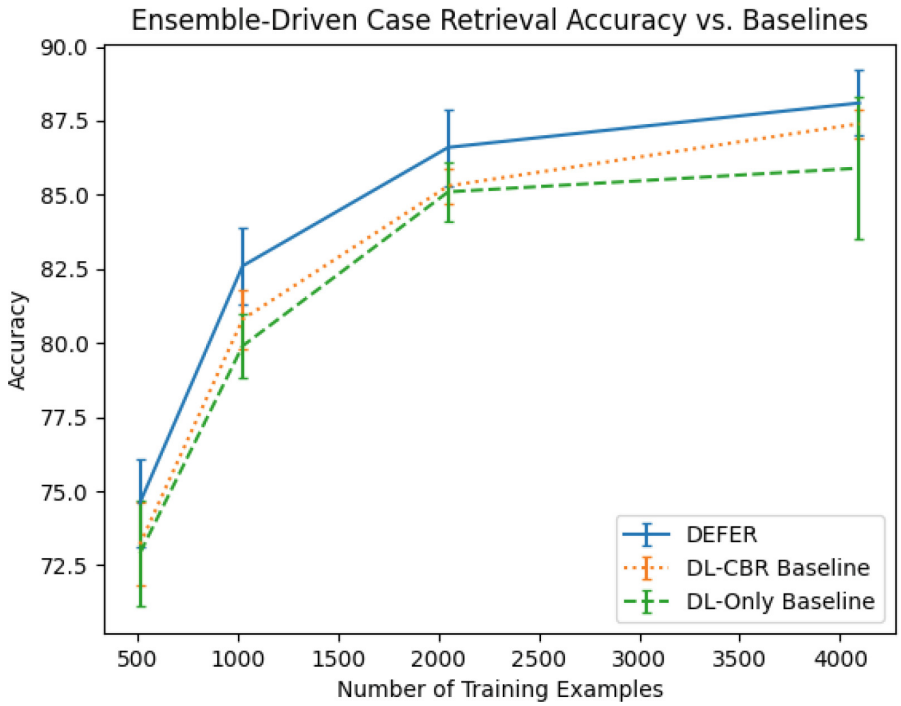

Fig. 3. Accuracy values for ensemble-driven case retrieval for different numbers of training examples. These are compared with our best-performing regular DL-CBR model [35] and the analogous DL-only model. Error bars represent one standard deviation.

from our prior work on ensembles of feature extractor models [21]. Experiments with larger numbers of training examples would be helpful to solidify the trends here, and we hypothesize that the larger number of network replicas in DEFER would benefit more from a larger training set than the DL-CBR baseline.

5.2 Tests of Hypothesis 2: Results from Ablation Studies

We conducted ablation studies to examine the influence of discriminator-based pruning, ratio-weighted replica voting, number of replicas, and value of n in the discriminator's top n predictions on model accuracy (Table 1). Taken together, these results support our second hypothesis, highlighting the benefit of our design decisions on DEFER's classification accuracy.

In these results, using the discriminator model to prune the search space significantly increases model accuracy. Intuitively, a larger n enables higher pruning accuracy for the discriminator, but based on these observations, either 1) the data imbalance created by relabeling the training data is still too significant, even with oversampling the positive examples, or 2) the "thinner" replica models, with fewer parameters, are more significantly impacted by the small-data nature of the experiment itself.

Ratio-weighted voting further increases the model accuracy, likely breaking arbitrarily-settled ties where two replicas (each sensitive to one of the discriminator's top-2 classes) clash over the ultimate prediction.

Table 1. Results for ablation studies, showing accuracy changes for DEFER when discriminator-based pruning and ratio-weighting are used (a), when the number of replicas is modified (b), and when the value of n for the top n discriminator predictions is modified (c). Best-performing models are boldfaced.

Parameters	% Accuracy	St. Dev.
No Discriminator	64.3	1.3
Discriminator Only	77.6	1.7
Ratio-Weighted Voting	**82.6**	**1.3**

(a)

# Replicas	% Accuracy	St. Dev.
2	82.4	1.4
5	82.4	1.2
10	**82.6**	**1.3**

(b)

Top n Classes	% Accuracy	St. Dev.
2	**82.6**	**1.3**
3	77.0	1.3
5	73.4	1.5

(c)

Finally, we consider the effects of different parameterizations for number of replicas and the value of n for the discriminator's top-n predictions used for pruning. The former subtly supports the intuitive trade-off between more replicas enabling replicas to be more specialized, but also leading to greater training data imbalance. The latter suggests that DEFER performs better when replicas have fewer degrees of freedom for voting; perhaps this is due to limited training data, in which case, additional experiments with more training examples may favor different values of n.

6 Conclusions

This paper presents DEFER, a new method that leverages the benefits of ensembles for feature extraction and case retrieval in an integrated DL-CBR classification system. DEFER uses ensembles of DL replicas sensitized to subsets of classes from the overall problem. Ideally, these replicas project a "superset" feature space that is approximated by a larger DL model into smaller subspaces that specialize in distinguishing between a few classes and all others, and reasoning over these localized feature spaces enables the CBR classifier to retrieve more accurately. This approach, combined with using a discriminator model to focus on a set of top-n classes and with weighting replica votes to reflect confidence in the retrieved case, leads to a novel DL-CBR integrated model that outperforms our previous best-performing approach and, interestingly, outperforms the analogous DL-only classifier.

Future work includes evaluation on additional datasets, including non-image data, and examining further variations of the number of training examples and features extracted. The experiments in this paper highlight that multi-network

approaches involve many interconnected parameters, providing opportunities for tuning. Additional design variations could include using the top-n positive classes for each replica (rather than the discriminator) for pruning, using various values of k for nearest-neighbor retrieval, extracting weights from the replica models to moderate replica voting, and/or concatenating the feature vectors extracted from each replica into a single vector for case retrieval. We expect that such refinements could further increase classification accuracy, as could the addition of case adaptation for full CBR.

This paper has only considered classification accuracy, but the DEFER approach might benefit interpretability as well. Traditional case-based classification is explained by the nearest case, or cases, according to a single feature vocabulary [16]. With DEFER, explanations can be offered by each replica to provide the user with explanatory cases from multiple feature perspectives. The effects of such richer explanations on user trust and satisfaction in classifications could be another interesting avenue to explore.

Acknowledgments. Work of the first three authors was funded by the US Department of Defense (Contract W52P1J2093009). We thank members of the DL-CBR research group at Indiana University for useful conversations.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/, software available from tensorflow.org
2. Amin, K., Kapetanakis, S., Althoff, K.-D., Dengel, A., Petridis, M.: Answering with cases: a CBR approach to deep learning. In: Cox, M.T., Funk, P., Begum, S. (eds.) ICCBR 2018. LNCS (LNAI), vol. 11156, pp. 15–27. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01081-2_2
3. Bach, K., Mork, P.: On the explanation of similarity for developing and deploying CBR systems. In: Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2020) (May 2020)
4. Barletta, R., Mark, W.: Explanation-based indexing of cases. In: Kolodner, J. (ed.) Proceedings of a Workshop on Case-Based Reasoning, pp. 50–60. DARPA, Morgan Kaufmann, Palo Alto (1988)
5. Barnett, A.J., et al.: Interpretable mammographic image classification using case-based reasoning and deep learning. In: Proceedings of IJCAI-21 Workshop on Deep Learning, Case-Based Reasoning, and AutoML (2021). https://arxiv.org/pdf/2107.05605
6. Bhatta, S., Goel, A.: Model-based learning of structural indices to design cases. In: Proceedings of the IJCAI-93 Workshop on Reuse of Design, pp. A1–A13. IJCAI, Chambery, France (1993)
7. Bonzano, A., Cunningham, P., Smyth, B.: Using introspective learning to improve retrieval in CBR: a case study in air traffic control. In: Leake, D.B., Plaza, E. (eds.) ICCBR 1997. LNCS, vol. 1266, pp. 291–302. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63233-6_500
8. Breiman, L.: Random forests. In: Machine Learning (2001)

9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artificial Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953, http://dx.doi.org/10.1613/jair.953
10. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: deep learning for interpretable image recognition. In: Advances in Neural Information Processing Systems 32, pp. 8930–8941. Curran (2019)
11. Cox, M., Ram, A.: Introspective multistrategy learning: on the construction of learning strategies. Artif. Intell. **112**(1–2), 1–55 (1999)
12. Cunningham, P., Zenobi, G.: Case representation issues for case-based reasoning from ensemble research. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 146–157. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44593-5_11
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
14. Domeshek, E.: Indexing stories as social advice. In: Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 16–21. AAAI Press, Menlo Park, CA (1991)
15. Fox, S., Leake, D.: Introspective reasoning for index refinement in case-based reasoning. J. Exper. Theoret. Artifi. Intell. **13**(1), 63–88 (2001)
16. Gates, L., Leake, D., Wilkerson, K.: Cases are king: a user study of case presentation to explain cbr decisions. In: Case-Based Reasoning Research and Development, ICCBR 2023, pp. 153–168. Springer (2023). https://doi.org/10.1007/978-3-031-40177-0_10
17. Hsieh, W.H., Shih, D.H., Shih, P.Y., Lin, S.B.: An ensemble classifier with case-based reasoning system for identifying internet addiction. Int. J. Environ. Res. Public Health **16**(7), 1233 (2019)
18. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2016). https://arxiv.org/abs/1608.06993
19. Kenny, E.M., Keane, M.T.: Twin-systems to explain artificial neural networks using case-based reasoning: comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (2019)
20. Leake, D.: An indexing vocabulary for case-based explanation. In: Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 10–15. AAAI Press, Menlo Park, CA (July 1991)
21. Leake, D., Wilkerson, Z., Crandall, D.: Extracting case indices from convolutional neural networks: a comparative study. In: Case-Based Reasoning Research and Development, ICCBR 2022 (2022)
22. Leake, D., Wilkerson, Z., Vats, V., Acharya, K., Crandall, D.: Examining the impact of network architecture on extracted feature quality for CBR. In: Case-Based Reasoning Research and Development, ICCBR 2023. Springer (2023). https://doi.org/10.1007/978-3-031-40177-0_1
23. Leake, D., Ye, X.: Harmonizing case retrieval and adaptation with alternating optimization. In: Case-Based Reasoning Research and Development - ICCBR 2021, pp. 125–139. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86957-1_9
24. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. ArXiv arXiv: abs/1710.04806 (2018)
25. Li, S., Harner, J.E., Adjeroh, D.A.: Random KNN. In: IEEE International Conference on Data Mining Workshops (2015)

26. Mathisen, B.M., Aamodt, A., Bach, K., Langseth, H.: Learning similarity measures from data. Progress Artifi. Intell. (2019)
27. Plaza, E., McGinty, L.: Distributed case-based reasoning. Knowl. Eng. Rev. **20**(3), 315–320 (2005)
28. Plaza, E., Ontañón, S.: Ensemble case-based reasoning: collaboration policies for multiagent cooperative CBR. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 437–451. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44593-5_31
29. Richter, M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S. (eds.) CBR Technology: From Foundations to Applications, chap. 1, pp. 1–15. Springer, Berlin (1998)
30. Sani, S., Wiratunga, N., Massie, S.: Learning deep features for knn-based human activity recognition. In: 25th International conference on Case-Based Reasoning (ICCBR 2017) (2017)
31. Schank, R., et al.: Towards a general content theory of indices. In: Proceedings of the 1990 AAAI spring symposium on Case-Based Reasoning. AAAI Press, Menlo Park, CA (1990)
32. Turner, J.T., Floyd, M.W., Gupta, K.M., Aha, D.W.: Novel object discovery using case-based reasoning and convolutional neural networks. In: Cox, M.T., Funk, P., Begum, S. (eds.) ICCBR 2018. LNCS (LNAI), vol. 11156, pp. 399–414. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01081-2_27
33. Turner, J.T., Floyd, M.W., Gupta, K., Oates, T.: NOD-CC: a hybrid CBR-CNN architecture for novel object discovery. In: Bach, K., Marling, C. (eds.) ICCBR 2019. LNCS (LNAI), vol. 11680, pp. 373–387. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29249-2_25
34. Wilkerson, Z., Leake, D., Crandall, D.: On combining knowledge-engineered and network-extracted features for retrieval. In: Case-Based Reasoning Research and Development, ICCBR 2021, pp. 248–262 (2021)
35. Wilkerson, Z., Leake, D., Vats, V., Crandall, D.: Extracting indexing features for CBR from deep neural networks: A transfer learning approach. In: Case-Based Reasoning Research and Development, ICCBR 2024. Springer (2024). https://doi.org/10.1007/978-3-031-63646-2_10
36. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. In: IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (2018)
37. Ye, X., Leake, D., Jalali, V., Crandall, D.J.: Learning adaptations for case-based classification: a neural network approach. In: Sánchez-Ruiz, A.A., Floyd, M.W. (eds.) ICCBR 2021. LNCS (LNAI), vol. 12877, pp. 279–293. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86957-1_19
38. Ye, X., Leake, D., Wang, Y., Zhao, Z., Crandall, D.: Towards network implementation of CBR: Case study of a neural network K-NN algorithm. In: Case-Based Reasoning Research and Development - 32nd International Conference, ICCBR-24. LNCS, vol. 14775, pp. 354–370. Springer (2024). https://doi.org/10.1007/978-3-031-63646-2_23