IEEE Open Journal of Signal Processing

Received 1 February 2023; revised 18 April 2023; accepted 22 April 2023. Date of publication 3 May 2023; date of current version 26 May 2023. The review of this article was arranged by Associate Editor Bin Li.

Digital Object Identifier 10.1109/OJSP.2023.3272281

Exploiting the Distortion-Semantic Interaction in Fisheye Data

KIRAN KOKILEPERSAUD [1] (Student Member, IEEE), MOHIT PRABHUSHANKAR [1] (Member, IEEE), YAVUZ YARICI [1], GHASSAN ALREGIB [1] (Fellow, IEEE), AND ARMIN PARCHAMI [2]

[1] OLIVES Lab at the Center for Signals and Information Processing, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA
[2] Ford Motor Company, Dearborn, MI 48126 USA

CORRESPONDING AUTHOR: KIRAN KOKILEPERSAUD (email: kpk6@gatech.edu).

This work was supported through the Ford-Georgia Tech Alliance Program.

ABSTRACT In this work, we present a methodology to shape a fisheye-specific representation space that reflects the interaction between distortion and semantic context present in this data modality. Fisheye data has the wider field of view advantage over other types of cameras, but this comes at the expense of high radial distortion. As a result, objects further from the center exhibit deformations that make it difficult for a model to identify their semantic context. While previous work has attempted architectural and training augmentation changes to alleviate this effect, no work has attempted to guide the model towards learning a representation space that reflects this interaction between distortion and semantic context inherent to fisheye data. We introduce an approach to exploit this relationship by first extracting distortion class labels based on an object's distance from the center of the image. We then shape a backbone's representation space with a weighted contrastive loss that constrains objects of the same semantic class and distortion class to be close to each other within a lower dimensional embedding space. This backbone trained with both semantic and distortion information is then fine-tuned within an object detection setting to empirically evaluate the quality of the learnt representation. We show this method leads to performance improvements by as much as 1.1% mean average precision over standard object detection strategies and .6% improvement over other state of the art representation learning approaches.

INDEX TERMS Contrastive learning, fisheye radial distortion, representation space, semantics and distortions.

I. INTRODUCTION

Autonomous vehicles (AV) have the potential to change existing transportation systems. However, one major concern is the interaction between their acquisition sensors (cameras) and their deep learning based decision algorithms. This concern exists because perception decisions made by an autonomous vehicle is dependent on the quality of the data they receive from the surrounding environment. In particular, camera setups with a wider field of view are attractive due to their ability to capture a more holistic representation of the entire scene. For this reason, fisheye camera lenses are gaining attention as the main vision sensor on these AV systems due to their effective receptive field of 180 degrees [1]. As a result of this advantage, fisheye cameras have seen widespread adoption in common vehicle settings such as parking assistance [2] and automated parking [3]. Despite their usage in diverse applications, fisheye cameras come with the unique challenge of exhibiting radial distortion as a function of distance from the center of the image. Analysis into the acquisition process of these cameras has shown that this distortion is an inherent consequence of projecting the hemispherical lens geometry onto a 2D plane [4]. A naive solution to this problem would be to simply apply a transformation that would rectify the distortion. However, it has been shown [5] that these types of approaches introduce artifacts at the edges and reduce the overall field of view of the image.

From a deep learning perspective, radial distortion introduces a plethora of issues because neural networks exhibit performance degradation outside of the pristine data setting [8]. Furthermore, most computer vision applications only

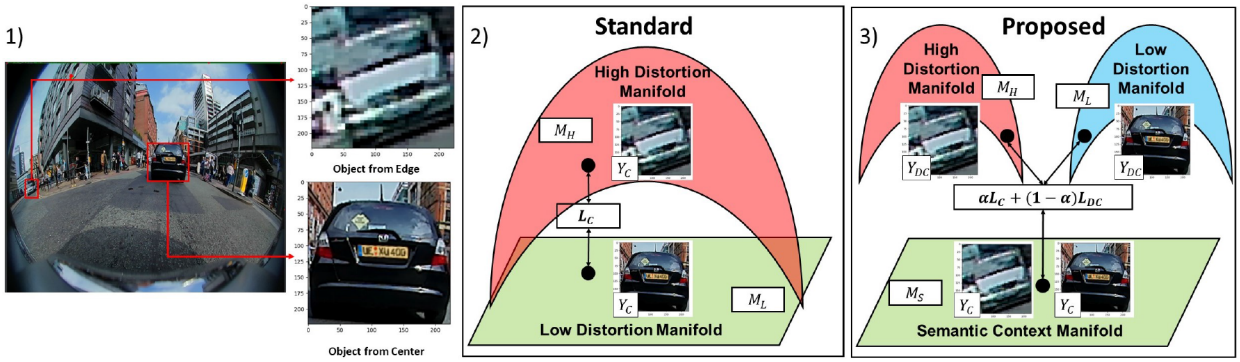

FIGURE 1. 1) This is an example fisheye object with associated car objects from the center and edge. 2) Previous methods view these objects as lying on separate manifolds that need to be corrected through a term that identifies them as belonging to the same semantic class L_C. 3) We propose an alternate view of the problem that views distortion and semantic context as belonging on separate sub-manifolds. The model is enabled to learn an intermediate representation that considers both concepts through a loss that enforces understanding of both the semantic class L_C and a distortion class L_{DC}.

consider data from narrow field of view cameras with mild radial distortion. As a result of these discrepancies, research has gone into developing deep learning approaches that maintain performance on fisheye data while simultaneously avoiding the sub-optimal process of rectifying the image. This research can be roughly divided into two sub-categories that we will refer to as model-centric and data-centric approaches. Model-centric refers to approaches [9], [10], [11] that attempt to change certain architectural features of a model with the intent of better conformation to an identified fisheye feature. Data-centric refers to approaches [12], [13] that manipulate the available training data in an attempt to better generalize on the fisheye setting. While these approaches have seen varied levels of success, they are lacking in the sense that they are very specifically optimized for their task of interest. In other words, these approaches introduce improvements within their target application (monocular depth estimation, semantic segmentation, etc.), but they do not identify core properties that a model's representation of data should have in order to be tuned to the fisheye setting.

In this work, we address this research gap by introducing a representation-centric approach specifically designed for a general fisheye paradigm. Our perspective on this paradigm can be understood by the example presented in Fig. 1. Within the first image, there are two cars: one that exists within the center of the frame and one that exists at the edge of the frame. It can be observed that the radial distortion property of fisheye data causes the car at the edge to exhibit a much higher level of distortion compared to the object taken from the center of the frame. To demonstrate the effect of this, we perform a toy experiment on the WoodScape [6] dataset to see how performance varies with respect to detecting center and edge located objects respectively. The results of this experiment on the Yolo v5 [7] architecture can be observed in Fig. 2 and it clearly shows a significant difference in performance of about. 06 mAP between objects found in the center compared to those located at the edge of the image. One interpretation, shown in Fig. 1, is that the distortion may cause the model

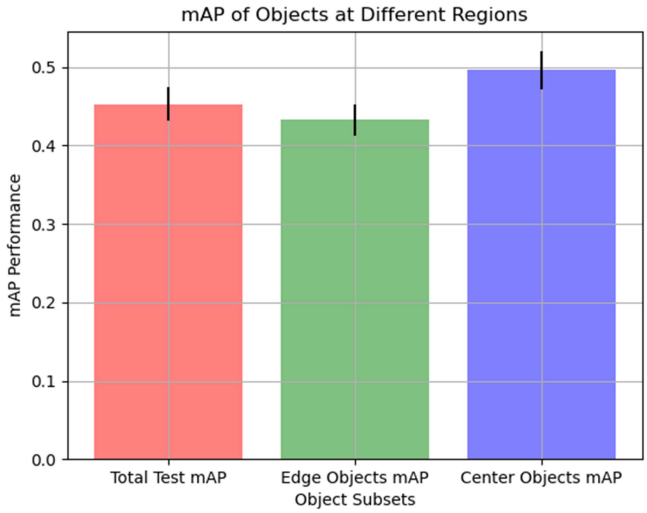

FIGURE 2. This shows the mean average precision(mAP) for a toy experiment on the WoodScape [6] dataset. An object detector is trained using the YOLO v5 [7] framework. Then, we compute total mAP, mAP of just the objects at the edge of the image, and mAP of objects within the center of the image.

to view these objects as coming from a high M_H and low M_L distortion manifold despite their membership within the same semantic class. While the data space reflects the distortion characteristic of the data, standard methods do not integrate distortion into the training paradigm and rely on just semantic labels Y_C with an associated semantic-based loss L_C. However, previous work [14], [15], [16] has shown that a model lacks generalization capability when the learnt representation does not reflect the underlying distribution of the data space. We argue in this work that the underlying distribution of fisheye data reflects not only semantic context or distortion alone, but a complex interaction between both. We visualize this perspective in part 3 of Fig. 1 where objects both exist on a semantic context manifold M_S as well as separate distortion specific sub-manifolds M_H and M_L. In this view of the problem, all

objects have a label with respect to the semantic manifold Y_C as well as labels that reflect their location within distortion space Y_{DC}. From this setup, it is then possible to train a model with a loss that integrates both the semantic characteristic L_C and distortion characteristic L_{DC} of the objects. We implement such a framework by first extracting objects from fisheye data and using their distance from the center to assign distortion based class labels alongside their semantic class label. We then take advantage of contrastive learning approaches [17] in order to explicitly enforce a model to learn a representation that reflects both the distortion and semantic characteristic of the object through a weighted contrastive loss $\alpha L_C + (1-\alpha)L_{DC}$. We then fine-tune the learnt representation within an object detection setting to empirically validate the approach. The target contributions of this work are:

1) We introduce a representation-centric approach to training fisheye data based on using contrastive learning as a way to constrain the interaction between semantics and distortion.
2) We perform an explicit analysis of this trade-off between being distortion-aware and semantically-aware within the context of an object detection setting.
3) We compare against standard object detection and representation learning baselines to demonstrate the advantage of our approach.

II. RELATED WORKS

A. RECTIFICATION APPROACHES

One application of deep learning on fisheye data involves developing models to rectify the fisheye data with the goal of removing distortion. [18] introduced the use of a CNN as a means to extract features, parse the surrounding scene, and estimate the distortion parameters necessary to rectify the image. [19] built upon to this idea to derive distortion parameters by enforcing that straight lines maintain their straightness. Recent work [20] has explored the usage of transformer architectures to model domain shifts across distortions. [21] demonstrated a self supervised approach for this task. While these works have shown good performance for the task of rectification, the downsides of a computational cost, rectification artifacts, and reduction in the field of view remain significant concerns.

B. MODEL-CENTRIC

Model-centric approaches refer to methods that introduce architectural changes in the hopes of performance improvements on their target task. Early work [22] made use of probabilistic appearance models for pedestrian tracking. [9] showed how changes to the output bounding box shape can improve the mIOU on distorted objects through better conformation to the distorted shape. Other work [10] has introduced new types of feature extraction strategies such as the usage of a hyperbolic convolutional kernel. [23] demonstrated the usage of adaptable deformable kernels for semantic segmentation. [24] introduced an additional feature pyramid block to detect smaller objects. Other ideas [11] showed how approximations to the domain of spherical data can work on fisheye data through the usage of an attention mechanism. While these architectural changes have shown performance improvements, it is unclear what aspects of fisheye data they are leveraging from the resultant features they extract. In our work, we make this explicit through our shaping of a representation space based on fisheye-centric principles of the interaction between distortion and semantics.

C. DATA CENTRIC

Data-centric approaches describe methods that intervene on the training data in the hopes of presenting the model with better views that are invariant to distortion. For example, [12] introduced a set of data augmentations that were tuned to the fisheye setting. Additionally, [13] showed how training with geometric perspectives can enable better training views within the context of a 3D object detection task. The main issue with these approaches is that these augmentations are tuned to a specific task and it isn't clear what general augmentation principle is at work. Our work circumvents this by introducing how to create a general representation principles for fisheye data.

D. CONTRASTIVE LEARNING

The main idea behind contrastive learning approaches is to learn a lower-dimensional embedding space where similar pairs of images (positives) project closer to each other than dissimilar pairs of images (negatives). The manner in which these positives and negatives are defined as well as their usage within the overall framework is what distinguishes contrastive learning methodologies from each other. Traditional approaches like [25], [26], [27] all choose positive instances by augmenting an image through some transformation and treating all other instances in the batch as the negative set. Within the domain of fisheye, [28] has utilized existing contrastive learning approaches on fisheye data for the task of semantic segmentation. This work differs fundamentally from ours in the sense that we are proposing a contrastive learning approach specifically geared towards creating a fisheye specific representation space, rather than a generic space based on previous learning approaches. In order to do this, we leverage the supervised contrastive loss [29] where positives and negative instances are chosen on the basis of belonging to the same semantic category or not. In other words, an additional constraint is placed on the embedding space to guide further understanding of learning similar and dissimilar data points. This inspired recent work [16], [30], [31] that generates labels using auxiliary information to shape a representation that is more appropriate for the application domain of medical and seismic data respectively. We introduce a way to make use of distortion auxiliary (distortion/semantic) information as a way to shape representations more suitable for the fisheye setting.

III. FISHEYE IMAGE ANALYSIS

A. DATASET

The fisheye dataset utilized in this paper is the WoodScape [6] autonomous driving dataset. This dataset is collected using four surrounding fisheye cameras on a moving vehicle over a variety of urban scenes. It has over 8.2 k images containing 5 different object categories: vehicle, pedestrian, bicycle, traffic light, and traffic sign. No public test set for this dataset is available, so the dataset was split into a training, validation, and test split of 80%, 10%, and 10% respectively. This specific dataset was chosen over the other fisheye datasets from [1] for the following reasons. Firstly, WoodScape is the only real-world fisheye AV dataset with object bounding box labels. Other fisheye AV datasets are either unlabeled with respect to object bounding boxes or are entirely simulated. The non-AV fisheye datasets that are both real and have bounding boxes are insufficient to demonstrate the interaction between distortion and semantic classes. This is because the AV setting has a wide distribution of objects across the entire image, while non-AV datasets only have center-focused objects. This makes it hard to study the impact of objects closer to the edge where the distortion is highest.

B. STATISTICS

In Fig. 2, we observe that the test set performance of objects located at the edge is lower than that of objects located in the center of the image. In this section, we build intuition regarding the interaction between object distortion and regional location. We provide analysis that demonstrate that the performance difference of Fig. 2 is due to the radial distortion of the fisheye images. To enable this analysis, we define two categories of objects - central and edge objects. All objects within an upper left image coordinate of (.25, .25) and lower right coordinate of (.75, .75) are central objects and objects outside this box are edge objects.

The first statistic we investigate is the distribution of the centered and edge objects across different classes in the training set. We can see from Fig. 3 that the majority of objects are located closer to the edge of the image. This plot was generated by extracting the center coordinate for every object in every class and then using our definition of center and edge to delegate which location bin they belong to. This indicates that in Fig. 2 the model was not biased towards detecting central objects better because of a greater prevalence of objects to train on from the center of the image. Another statistic to investigate is the size of objects at different regions of the image. This is shown in Fig. 4 for the three most prevalent classes: vehicles, pedestrians, and bicycles. In this plot, every object's distance from the center is plotted against the area of their respective bounding box. We observe that the vast majority of objects in every class maintains a roughly similar size regardless of their distance from the center. This is further validated by the histogram in Fig. 5 that shows the number of objects as a function of the object area for the majority classes. This histogram shows that most objects within each

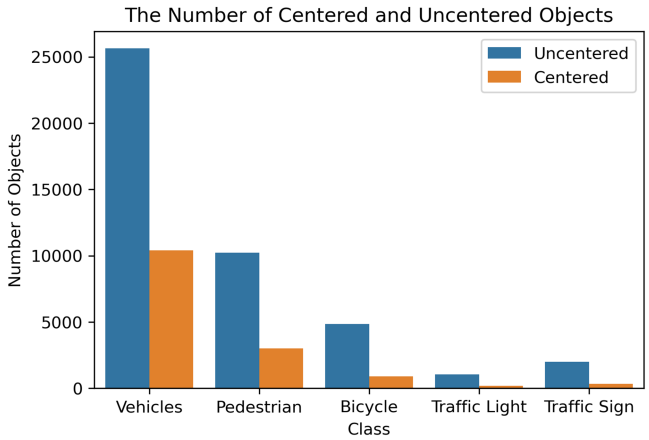

FIGURE 3. This plot shows the number of objects located at the edge and the center for each of the classes in the WoodScape dataset.

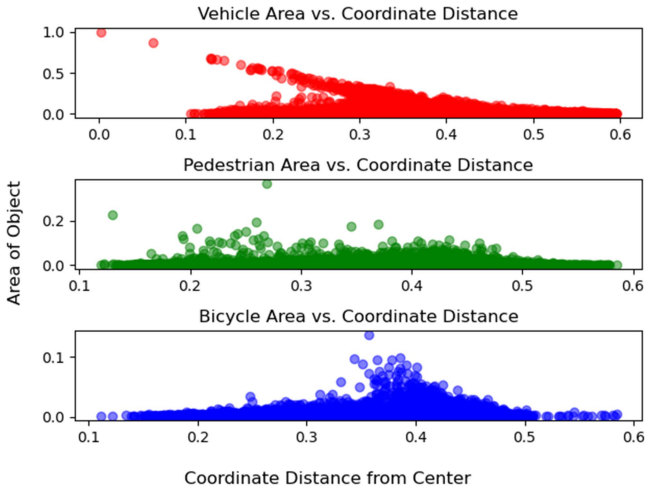

FIGURE 4. This plot shows the relationship between the coordinate distance and object area for every object among the three most prevalent classes in the WoodScape dataset. Distance was computed as the mean squared error between the center coordinate of the image (.5,.5) and the object's center coordinate. Object area was computed by multiplying the height and width of the object's bounding box. The maximum possible distance is 0.707 and the lowest is 0.

class have a similar size to each other. Together these plots show that the worse performance on the edges is not due to having smaller sized objects to detect compared to the center. Both regions have similarly sized objects with the difference being the higher radial distortion exhibited by the objects at the edge.

To further build on this idea, we attempt various ways in Fig. 6 to quantify the distortion exhibited by objects on the edge in comparison to objects in the center. The first plot shows how distortion changes according to the radial distortion mathematical model described in [4]. This shows the associated distortion should theoretically increase the further the away from the center of the image associated features are located. To evaluate this empirically, we also extract each

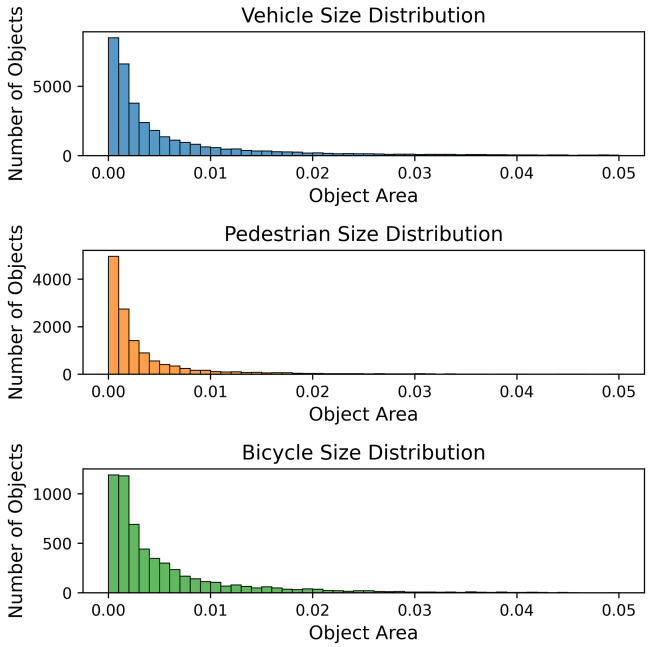

FIGURE 5. **This is a histogram of the number of objects at different sizes for each class. The area for each object is computed by multiplying the height and width for each objects bounding box coordinates. This is then ranked and binned to create this histogram.**

object from both the pedestrian and bicycle classes and assign them as center and edge objects according to the conventions we introduce. We then compute BRISQUE [32] features that quantifies losses in "naturalness" due to distortions. This results in every image having an associated 36×1 feature vector. We compute a single value by averaging across this feature vector for each object. We compute the mean and standard deviation for this value across all objects in each class and associated regional location and plot the associated Gaussian. We observe that there is significant separation in BRISQUE features between the edge and center located objects. Specifically, this corresponds to a percent overlap of 18.55% for the pedestrian class 10.90% for the bicycle class which indicates that the distortion had some effect on the distributions of these objects.

IV. METHODOLOGY

Our methodology follows three distinct steps: regional label extraction, followed by pre-training of a ResNet-18 network [33] with a linear combination of contrastive losses, and finally fine-tuning the learnt representation with an object detection head. The overall philosophy behind this approach is to enable the model to recognize both semantic and distortion related information within its representation space. The regional extraction provides us with the labels for distortion and the contrastive learning operates with a weighted loss that constrains the representations learnt in terms of both semantic and distortion related contexts.

A. REGIONAL CLASS LABEL EXTRACTION

In order to train a model to recognize both semantic and distortion concepts, we need to acquire labels that reflect both. In the case of semantic information, the label files for every image identifies the class that each object belongs to. However, there isn't an explicit label that reflects the distorted nature of the object. In order to acquire this, we use the bounding box information of each object to receive the center coordinate of the object. This coordinate is important because the further the object from the center of the frame, the greater the distortion characteristics it exhibits. Therefore, it is necessary to define a threshold by which all objects outside of this threshold belong to a high-distortion class and all objects within this threshold belong to a lower distortion class.

With these considerations in mind, the process to define these distortion-based labels is detailed in Fig. 7. For each image in the training set, we extract each individual object as its own image patch along with its associated class label and bounding box coordinates. Every object is immediately assigned its original class label, but it is also assigned an additional label to describe its distortion class. This is done by analyzing the center coordinate. If the center coordinate of the object belongs in the inscribed box with an upper left coordinate of (.25, .25) and lower right coordinate of (.75, .75) then we consider the object to belong close to the center of the image. In this case, we assign the object with the additional label of a lower distortion version of its class. If the center coordinates lies outside of this defined box, we assign the object with the label of a higher level distortion version of its class. This extraction and label assignment process is repeated across the training set to create a large pool of object patches with associated label information. For example, this means that every car object receives its semantic class of car as well as its appointment as a high or low distortion version of its class, such as highly distorted car and barely distorted car, resulting in 10 possible distortion classes due to two variants of each of the 5 classes.

B. CONTRASTIVE PRE-TRAINING

After disentangling the training set into object patches labeled with both a semantic and distortion class, we perform a contrastive learning objective that constrains the representation to consider both concepts. The overall block diagram of the proposed method is summarized in Fig. 8. Given an input batch of extracted objects x_k, associated distortion label (y_{dk}), and associated class label (y_{ck}) to form the triplet $(x_k, y_{dk}, y_{ck})_{k=1,\ldots,N}$, we perform augmentations on the batch twice in order to get two copies of the original batch with $2\,N$ object patches and corresponding labels. These augmentations are random resize crop, random horizontal flips, random color jitter, and data normalization. This process produces a larger set $(x_l, y_{dl}, y_{cl})_{l=1,\ldots,2\,N}$ that consists of two versions of each object patch that differ only due to the random nature of the applied augmentation. Thus, for every object patch x_k, distortion label y_{dk}, and class label y_{ck} there exists two views

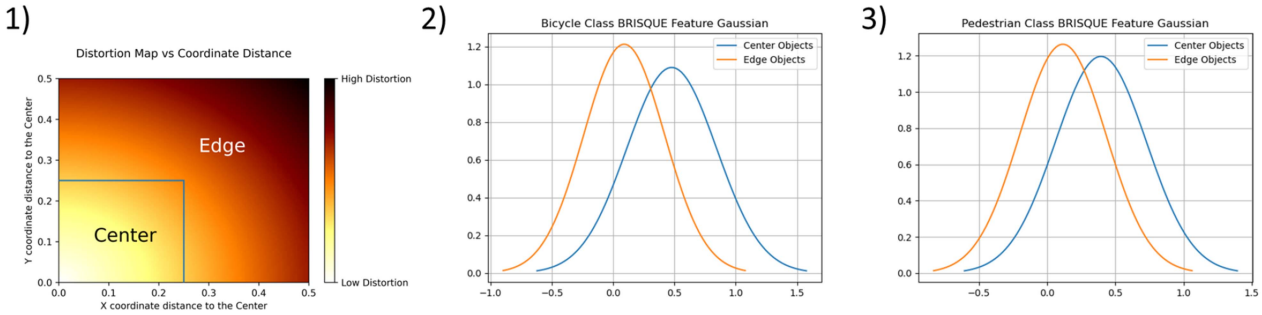

FIGURE 6. This plot shows different statistics regarding distortion of center and edge objects. 1) This shows how distortion varies according to the fisheye polynomial distortion model $d(\rho) = a_0 + a_2\rho^2 + a_3\rho^3 + a_4\rho^4$ where $\rho = \sqrt{(x^2 + y^2}$ and $d(\rho)$ represents the associated distortion at ρ distance. The parameters $a_0 \ldots a_4$ are chosen from calibration files provided by the WoodScape dataset. We also show what x and y coordinate distances correspond to our definition of center and edge. 2) and 3) To produce these plots, every object for both the bicycle and pedestrian classes were passed through the BRISQUE [32] algorithm to produce a 36 x 1 feature vector that. The mean and standard deviation of this vector were takent to produce the associated Gaussian.

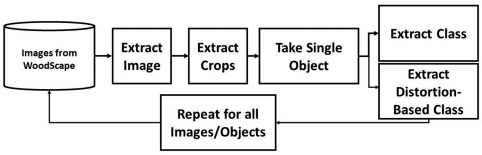

FIGURE 7. This figure details the process by which objects are extracted and assigned an additional label based on distortion information. Objects from the WoodScape dataset are extracted and assigned their class label as well as a distortion class label based on whether the object belongs in the defined center region box or not. This is repeated for all objects to mine a training set of object patches with both semantic and distortion labels.

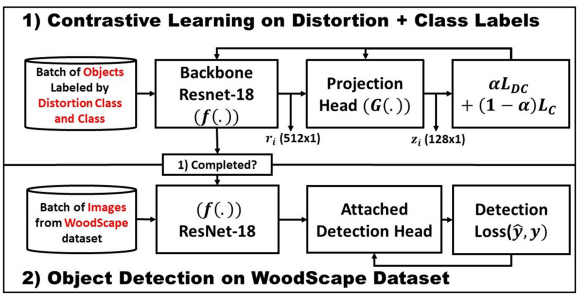

FIGURE 8. This is the overall setup. 1) The backbone ResNet-18 model is trained using the combined contrastive loss on objects that incorporate both distortion and semantic class labels. 2) The model pre-trained with the contrastive loss in the previous step is transferred as the backbone for a YOLO v5 object detection setup.

of the image $x_{2\,k}$ and x_{2k-1} and two copies of the labels that are equivalent to each other: $y_{2dk-1} = y_{2dk} = y_{dk}$ and $y_{2ck-1} = y_{2ck} = y_{ck}$.

From this point, we perform the first step in Fig. 8, where a linear combination of supervised contrastive losses is performed on the identified distortion and class based labels. The labeled augmented batch of object patches is forward-propagated through an encoder network $f(\cdot)$ that we set to be the ResNet-18 architecture [33]. This results in a 512-dimensional vector r_i that is sent through a projection network $G(\cdot)$, which further compresses the representation to a 128-dimensional embedding vector z_i. $G(\cdot)$ is chosen to be a multi-layer perceptron network with a single hidden layer. This projection network is utilized only to reduce the dimensionality of the embedding before computing the loss and is discarded after training. A supervised contrastive loss is performed on the output of the projection network in order to train the encoder network to have a weighted constraint based on both class and distortion labels. In this case, embeddings with the same class label are enforced to be projected closer to each other while embeddings with differing class labels are projected away from each other. At the same time, another loss enforces embeddings with the same distortion label to be projected closer to each other while embeddings with differing distortion labels are projected away from each other. This results in a class based supervised contrastive loss L_C and a distortion class based supervised contrastive loss L_{DC}. The form of the distortion contrastive loss is shown as: $L_{DC} = \sum_{i \in I} \frac{-1}{|DC(i)|} \sum_{dc \in DC(i)} log \frac{exp(z_i \cdot z_{dc}/\tau)}{\sum_{a \in A(i)} exp(z_i \cdot z_a/\tau)}$ where i is the index for the object patch of interest x_i. All positives dc for object patch x_i are obtained from the set $DC(i)$ and all positive and negative instances a are obtained from the set $A(i)$. Set $DC(i)$ represents all other object patches in the batch with the same distortion class label dc as the object patch of interest x_i while set $A(i)$ refers to every other element in the same batch as x_i. Additionally, z_i is the l2-normalized embedding for the object patch of interest. z_{dc} represents the embedding for the distortion class positives, and z_a represents the embeddings for all positive and negative instances in the set $A(i)$. τ is a temperature scaling parameter that is set to. 07 for all experiments. The loss function operates in the embedding space where the goal is to maximize the cosine similarity between embedding z_i and its set of distortion class positives z_{dc}. The class based contrastive loss follows the same setup except positives are chosen on the basis of only class information. In order to weight the influence that each term has on shaping the representation learnt by the model we introduce an α parameter. This weighting between each contrastive loss can be represented by: $L_{total} = \alpha L_{DC} + (1 - \alpha)L_C$. In this way,

we are creating a linear combination of losses from different label distributions for the same object patch.

C. OBJECT DETECTION FINE-TUNING

After training the encoder on objects with a combined contrastive loss, we move to the second step in Fig. 8 where the weights of the encoder are transferred into the backbone of the object detection setup and a YOLO v5 object detection head [7] is appended to the output of the encoder. The images from the WoodScape training set are input into this setup and the model is trained to perform standard object detection. In this way, we leverage knowledge learnt from constraining representations based on class and distortion information in order to improve performance for the task of object detection.

V. RESULTS

A. TRAINING DETAILS

The hyperparameters utilized can be divided into those applied for the contrastive learning step and that meant for the object detection fine-tuning. During contrastive learning, we set a batch size of 64 and training was performed for 25 epochs. A stochastic gradient descent optimizer was used for contrastive pre-training with a learning rate of. 001, weight decay of. 0001, and momentum of. 9. The applied augmentations are random resize crop, random horizontal flips, random color jitter, and data normalization to the mean and standard deviation of the Woodscape dataset. The comparison methods of SimCLR [25], Moco v2 [26], and PCL [27] were trained in the same manner with certain hyper-parameters specific to each method. Specifically, Moco v2 was set to its default queue size of 65536. Additionally, PCL has hyper-parameters specific to its clustering step, but the original documentation made these parameters specific to the Imagenet [34] dataset on which it was originally built for. To fit these parameters to our setting, the clustering step was reduced in size.

Object detection fine-tuning on top of the contrastively trained representation space follows many of the same hyper-parameter choices as in the original Yolo v5 training setup. The main points to note are resizing all images to a size of 640 x 640, a training time of 100 epochs, a chosen batch size of 32, and a stochastic gradient descent optimizer with a learning rate of. 01 that follows a cosine learning rate scheduler as training progresses. Further details of the architecture of the object detection head and its parameters can be found in the original YOLO v5 codebase [7].

B. ALPHA PARAMETER ANALYSIS

In order to get a sense of the trade-off between optimizing for distortion and semantic information, we vary the alpha parameter on the combined contrastive loss $L_{total} = \alpha L_{DC} + (1-\alpha)L_C$ that we introduce. In this way, we can observe how shaping the representation with respect to each loss term effects downstream performance once fine-tuning for the object detection task is performed. We observe in Fig. 10 that the choice of alpha has a significant impact on the downstream performance of the model. Specifically, we note that when $\alpha = .5$, the performance is highest and when $\alpha = 0$ or $\alpha = 1$ the performance is no better than the standard object detection baseline. In other words, when the learnt representation is forced to consider both the distortion and semantic information equally, the performance is much better than when the model is subjected to each alone. This trend holds as performance decreases on either side of $\alpha = .5$ as the α value increases or decreases. This result is significant because it validates the idea that a good representation space for fisheye data is one that reflects this interaction between distortion and semantic information. In particular, an $\alpha = .5$ yielding the best performance suggests that both are equally important for shaping a good representation space. A possible reason for this performance increase relates to the work of [35] where the authors show that a representation space should not map all instances of a class to the same point, but rather uniformly distribute them across the surface of a hypersphere based on implicit sub-classes within each higher order class. Within our setting this means that not only should objects of the same class map close to each other, but also be distributed based on distortion characteristics. We also observe visually in Fig. 9 that our method is able to more accurately detect objects compared to the standard baseline method. Additionally, we see a performance boost for both center and edge mAP performance in Table 1.

We also compare our approach at $\alpha = .5$ to other representation approaches in Table 2. Our approach beats all other contrastive learning strategies that only consider a simple augmentation as a means of constraining the representation space.

TABLE 1. We Observe a Higher Performance on the Edge and Center Objects Compared to the Baseline Approach

Edge and Center Study			
Method	Total mAP	Edge mAP@.6	Center mAP@.6
Baseline Yolo v5	.453	.433	.496
Ours (alpha = .5)	**.464**	**.445**	**.505**

The bold values indicate the highest performing in each column.

TABLE 2. This Shows the Performance of Our Best Contrastive Pre-Training Model Against Other Well Known Representation Learning Approaches. Performance is Reported At Different IOU Thresholds for the mAP Metric

Comparison with other Representation Learning Approaches			
Method	mAP@.5	mAP@.6	mAP@.7
Standarad OD	.457	.454	.433
PCL [27]	.458	.457	.441
SimCLR [25]	.453	.451	.430
Moco v2 [26]	.462	.459	.444
Ours (α=.5)	**.467**	**.464**	**.448**

The bold values indicate the highest performing in each column.

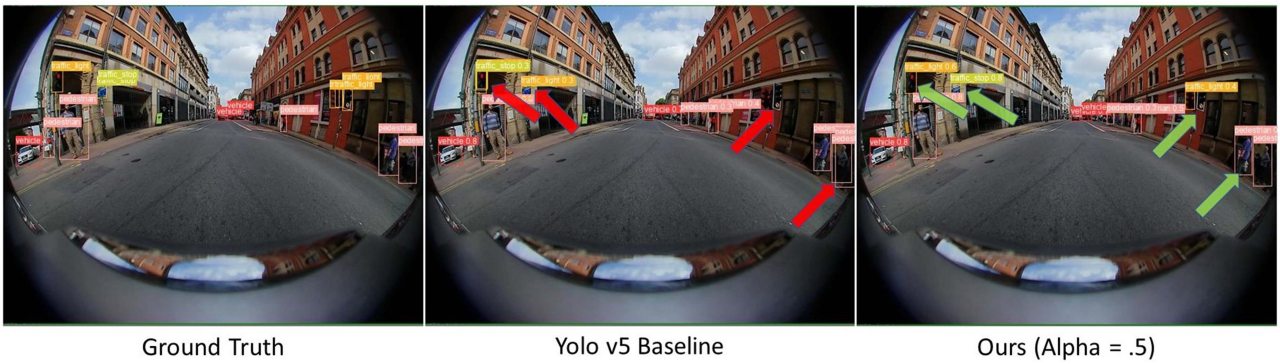

FIGURE 9. This is a visual comparison between objects detected by the Yolo v5 baseline architecture and our method with an alpha parameter of. 5. We include red and green arrows to highlight where our method did better than the baseline Yolov5.

TABLE 3. We Perform an Ablation Study of Varying the Definition of the Boundary Between Center and Edge Objects. Standard Box Refers to the Size We Use in the Paper of an Upper Left Coordinate of (.25,.25) and Lower Right Coordinate of (.75,.75). Large Box Refers to an edge/center Boundary Box Defined by the Coordinate Pair (.1,.1) and (.9,.9). Small Box Efers to an edge/center Boundary Box Defined by the Coordinate Pair (.33,.33) and (.66,.66)

Comparison with Different Edge/Center Box Definitions	
Box Definition	mAP@.6
Standard Box	.464
Large Box	.431
Small Box	.424

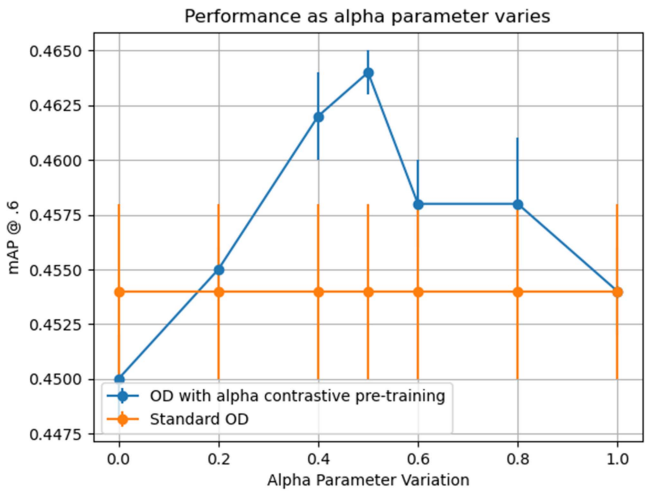

FIGURE 10. The plot shows the effect of varying the alpha parameter on the combined contrastive loss. We compare this against a standard object detection (Standard OD) setup that does not benefit from access to contrastive pre-training of the backbone network.

We also note that this improvement in performance is consistent across different bounding box intersection over union (IOU) thresholds. This indicates that the resultant model produces higher quality bounding boxes compared to the other methods. Part of the reason for this improvement is that traditional approaches choose positives and negatives that are less reflective of the data distribution. For example, by choosing the only positive pair as an augmentation and every other point in the batch as a negative, this leads to situations where the negatives consist of points that should be positives. This is avoided in our approach through having to a much more diverse pool of positives based on both class and distortion related considerations.

C. EXPERIMENT VARIATION STUDIES

We also study different ways in which to define the distorted and clean image regions, different architectures, and different chosen contrastive learning hyperparameters. Within our work so far, we have defined a box by which all objects inside this box are considered low distortion and all objects outside this box are considered high distortion labels. In Table 3 we explore the effect of a larger and smaller definition of this boundary. We observe in both cases performance degrades compared to our standard box based on the midpoint distance between the center and upper left corner of the image. Another way of defining these regions is to discretize the distance from the center as a series of ranges by which each range of values would denote a separate distortion class. To study this possibility, we discretize the distance from the center into l different distortion levels and train the contrastive learning setup in the same way discussed previously with the difference being the addition of additional distortion classes for every semantic class. We then fine-tune the representation trained in this manner for the task of object detection on the WoodScape dataset and report the results in Fig. 11. We observe that any level of discretization beyond the low and high levels we introduced leads to a substantial drop-off in performance in terms of mAP. Part of the reason for this is that distortion in fisheye data doesn't change at a fast enough rate that would cause substantial differences between distortion levels. Therefore, from a contrastive learning point of view, the loss does not have significant enough differences in terms of features to contrast between.

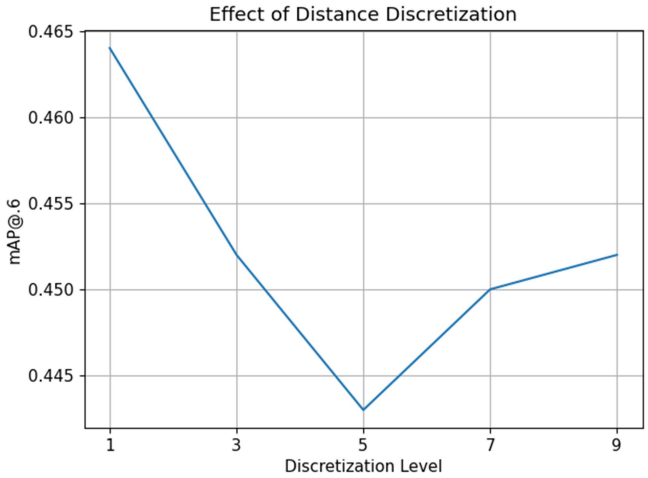

FIGURE 11. In this plot, the definition of objects that exhibit high and low distortion are changed to include multiple possible regions of distortion. Discretization level indicates the number of distortion regions generated. The performance shown is how well each contrastive pre-training did when the distortion class is re-defined with the larger number of distortion levels.

TABLE 4. We Study the Effect of Variation of Batch Size With a Fixed Temperature Of .07 as Well as Variation of Temperature With a Fixed Batch Size of 64. Our Chosen Parameters of a Batch Size of 64 and Temperature Of .07 Lead to the Best Performance

Contrastive Learning Hyperparameter Study			
Batch Size	mAP@.6	Temperature	mAP@.6
32	.457	.07	.464
64	.464	0.2	.458
128	.458	0.5	.449

TABLE 5. We Study the Effect of Training on Different Families of Architectures Under the Baseline Method and Ours With the Best Empirical Alpha Parameter

Architecture Study			
Method	Yolo v5 [7]	RetinaNet [36]	EfficientDet [37]
Baseline	.453	.229	.241
Ours (alpha = .5)	**.464**	**.250**	**.251**

The bold values indicate the highest performing in each column.

We also ensure that our methods works in other object detection frameworks. We show a performance improvement over baseline RetinaNet [36] and EfficientDet [37] in Table 5 when integrating our approach into the backbone pre-training of each network with an alpha weighting of .5. Additionally, we analyze the sensitivity of our approach with respect to batch size and temperature scaling in Table 4. As expected a higher temperature leads to poorer performance, as observed in [29], and an optimal batch size choice results in the best performing setting.

VI. CONCLUSION

In this work, we investigate how a contrastive learning methodology can be used to enforce a model's representation space to reflect the distortion and semantic interaction inherent within fisheye data. We show how our method reflects this interaction through experiments that vary the alpha parameter during contrastive pre-training. Additionally, further experiments that compare against different representation learning strategies and discretization levels shows the introduced strategy out-performs existing approaches as well as standard object detection. We conclude from these experiments that a quality representation space is one the reflects the features of the data on which it is trained on. In this case, our models are better able to overcome the fisheye radial distortion by being allowed to integrate this information within the training process.

REFERENCES

[1] V. R. Kumar, C. Eising, C. Witt, and S. Yogamani, "Surround-view fisheye camera perception for automated driving: Overview, survey and challenges," *IEEE Trans. Intell. Transp. Syst.*, 2023.
[2] C. M. Hughes, M. Glavin, E. Jones, and P. Denny, "Wide-angle camera technology for automotive applications: A review," *IET Intell. Transport Syst.*, vol. 3, no. 1, pp. 19–31, 2009.
[3] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," *Image Vis. Comput.*, vol. 68, pp. 88–101, 2017.
[4] K. Miyamoto, "Fish eye lens," *J. Opt. Soc. Amer.*, vol. 54, no. 8, pp. 1060–1061, 1964.
[5] V. R. Kumar, S. Yogamani, M. Bach, C. Witt, S. Milz, and P. Mäder, "Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8177–8183.
[6] S. Yogamani et al., "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9308–9318.
[7] G. Jocher et al., "ultralytics/yolov5: V7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022.
[8] D. Temel, G. Kwon, M. Prabhushankar, and G. AlRegib, "Cure-TSR: Challenging unreal and real environments for traffic sign recognition," 2017, *arXiv:1712.02463*.
[9] H. Rashed et al., "FisheyeYOLO: Object detection on fisheye cameras for autonomous driving," in *Proc. Mach. Learn. Auton. Driving NeurIPS 2020 Virtual Workshop*, 2020, vol. 8.
[10] O. Ahmad and F. Lecue, "Fisheyehdk: Hyperbolic deformable kernel learning for ultra-wide field-of-view image recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 6, pp. 5968–5975.
[11] J. Miao, Y. Liu, J. Liu, A. Argyriou, Z. Xu, and Y. Han, "Improved face detector on fisheye images via spherical-domain attention," in *Proc. IEEE Symp. Comput. Commun.*, 2021, pp. 1–7.
[12] G. Blott, M. Takami, and C. Heipke, "Semantic segmentation of fisheye images," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.
[13] E. Plaut, E. B. Yaacov, and B. E. Shlomo, "3D object detection from a single fisheye image without a single fisheye training image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3659–3667.
[14] M. Prabhushankar and G. AlRegib, "Introspective learning: A two-stage approach for inference in neural networks," *Adv. Neural Inf. Process. Syst.*, 2022.
[15] G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated gradient representations for anomaly detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 206–226.
[16] K. Kokilepersaud, M. Prabhushankar, G. AlRegib, S. T. Corona, and C. Wykoff, "Gradient-based severity labeling for biomarker classification in oct," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 3416–3420.
[17] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, vol. 8, pp. 193907–193934, 2020.

[18] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao, "Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 469–484.
[19] Z. Xue, N. Xue, G. S. Xia, and W. Shen, "Learning to calibrate straight lines for fisheye image rectification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1643–1651.
[20] S. Yang, C. Lin, K. Liao, and Y. Zhao, "Fishformer: Annulus slicing-based transformer for fisheye rectification with efficacy domain exploration," 2022, *arXiv:2207.01925*.
[21] C. H. Chao, P. L. Hsu, H. Y. Lee, and Y. C. F. Wang, "Self-supervised deep learning for fisheye image rectification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 2248–2252.
[22] M. Saito, K. Kitaguchi, G. Kimura, and M. Hashimoto, "People detection and tracking from fish-eye image based on probabilistic appearance model," in *Proc. IEEE SICE Annu. Conf.*, 2011, pp. 435–440.
[23] C. Playout, O. Ahmad, F. Lecue, and F. Cheriet, "Adaptable deformable convolutions for semantic segmentation of fisheye images in autonomous driving systems," 2021, *arXiv:2102.10191*.
[24] P. Y. Chen, J. W. Hsieh, M. Gochoo, C. Y. Wang, and H. Y. M. Liao, "Smaller object detection for real-time embedded traffic flow estimation using fish-eye cameras," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 2956–2960.
[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[26] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.
[27] J. Li, P. Zhou, C. Xiong, and S. CH Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Representations*, 2020.
[28] C. Ramchandra et al., "Fisheyepixpro: Self-supervised pretraining using fisheye images for semantic segmentation," *Electron. Imag.*, vol. 34, pp. 1–6, 2022.
[29] P. Khosla et al., "Supervised contrastive learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 18661–18673, 2020.
[30] K. Kokilepersaud, M. Prabhushankar, and G. AlRegib, "Clinical contrastive learning for biomarker detection," 2022, *arXiv:2211.05092*.
[31] K. Kokilepersaud, M. Prabhushankar, and G. AlRegib, "Volumetric supervised contrastive learning for seismic semantic segmentation," in *Proc. 2nd Int. Meeting Appl. Geosci. Energy,*2022, pp. 1699–1703.
[32] A. Mittal, K. A. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[34] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[35] D. Y Fu, M. F. Chen, M. Zhang, K. Fatahalian, and C. Ré, "The details matter: Preventing class collapse in supervised contrastive learning," in *Proc. Comput. Sci. Math. Forum*, 2022, vol. 3, Art. no. 4.
[36] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ""Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[37] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10781–10790.