Egocentric Data as Natural Adversarial Examples

Mary Williamson
Stanford University (SCPD)
marywill@stanford.edu

Abstract

Natural adversarial examples [7] have been shown to significantly degrade neural classifier performance and that today's models have common failure modes such as an over-reliance on background, texture, and spurious correlations in the training data. At the same time, developments in video and augmented reality are providing new sources of images with significantly different distributions. The recently released Ego4D video dataset contains over 3,000 hours of video from almost 1,000 volunteers who wore cameras and performed daily activities. I use a subset of that data, from their Hands & Objects Benchmark, which contains annotations for static frames containing objects. I evaluate five popular model families, including ResNet, ConvNext, and Vision Transformer, zero shot on this data and after finetuning. Zero shot Top-1 accuracy across all models is very low on this Ego4D Subset with the highest at 18%. After finetuning, performance improves but is still low, with the highest Top-1 accuracy achieved at 38% by a ViT. This demonstrates further weaknesses in the generalization ability of today's neural classifiers and the potential of egocentric data from augmented reality applications to improve the robustness of these models.

A. Introduction

Several works have demonstrated that ImageNet [1] has relatively simple test examples [14], that today's neural classification models are not robust to distributional shifts [7], and that these models rely on spurious [3] (e.g. planes against a blue sky) or textural cues too much [4]. Consequently, both artificial and naturally-selected adversarial examples [7] have proven effective at degrading neural classifier performance. At the same time, developments in video research and applications for augmented reality are providing new sources of images and video with significantly different distributions. The Ego4D dataset [5] is a recently released large video dataset of egocentric video, which presents an opportunity to study out of domain generalization and robustness with natural examples of a highly different distribution.

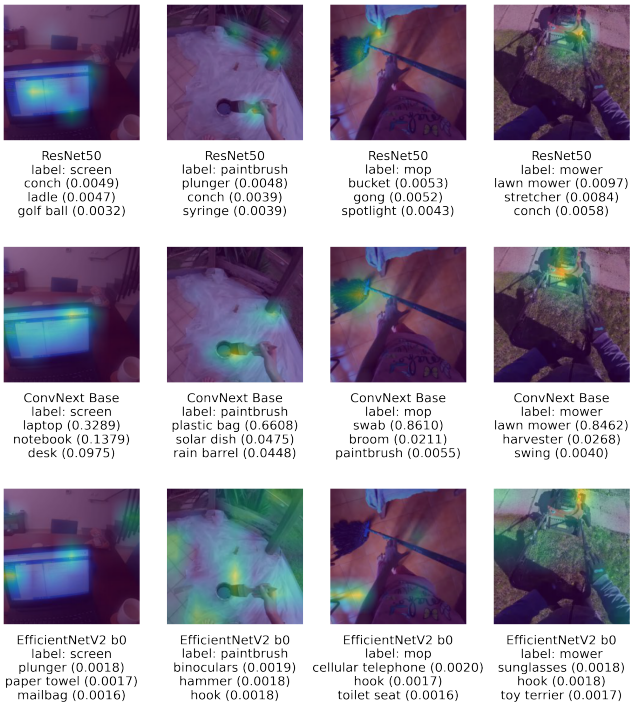

Figure 1. Example images from the Ego4D Subset introduced in this work demonstrate that egocentric data are natural adversarial examples for popular neural classifiers. From left, the Ego4D annotation "screen" versus laptop demonstrates the difficulty of mapping one label space to another. The paintbrush in the second image is relatively small, and all models fail to classify it. The mop is viewed from an unusual perspective due to the camera's location on the body. 2 out of 3 models actually classify the lawn mower correctly. The overlay is gradient-weighted Class Activation Mapping (GradCAM [16]).

In recent years, the robustness of neural classification models has been tested and found lacking. Models have been found to engage in "shortcut learning" [3], where they rely on background, texture, pose or other cues instead of true image understanding. Related to this lack of real-world understanding in learned representations are natural adver-

sarial examples. These are human-produced images, as opposed to artificially created data, which induce common models to make incorrect predictions. The ImageNet-A and ImageNet-O datasets [7] contain images filtered by hand which cause common models to generate incorrect predictions with high confidence. In that work, the authors evaluate a range of models and find that all perform very poorly on their datasets and that popular models contain similar failure modes.

The ImageNet-A dataset contains highly curated images adversarially filtered for poor performance by common models. On the other hand, egocentric data like that of the Ego4D dataset provides the opportunity to study truly "natural" examples from a different modality and perspective due to the camera typically being worn on the head. Ego4D has several subsets of data, but in this project, I use the Hands & Object benchmark, which contains video footage of participants manipulating objects with their hands. I filter this video data for annotated static images of objects. In addition, I further remove any images missing an annotation which maps to an ImageNet class and any images which contain more than one annotated ImageNet class. This results in an ultimate test set of 3,321 images for ImageNet-1K evaluation and 7,398 for ImageNet-21K evaluation [1]. This data has 69 unique ImageNet-1K classes and 248 ImageNet-21K classes using the above methodology. Hereafter, I refer to this data as the Ego4D Subset.

I evaluate several common neural classifiers zero shot accuracy on this new data. I evaluate both convolutional models and Vision Transformers of various sizes. Model families used include: ResNet [6], RegNetY [13], ConvNext [11], Vision Transformer [2], and EfficientNet V2 [18] [19]. I find that all models regardless of architecture perform very poorly without additional finetuning, though larger models have some benefit. I also evaluate models pretrained on the larger ImageNet-21K dataset and then finetuned on ImageNet-1K, which show a small benefit. The last zero shot evaluation I perform is on two models pretrained solely on ImageNet-21K. For this evaluation, the Ego4D Subset annotations are re-mapped to the larger 21,843 class set of ImageNet-21K, and I use a multilabel classification variant I developed due to annotation ambiguity. See the Data section for further discussion. I also finetune one model from each of the model families on half of the new Ego4D Subset, and while performance on the Subset improves, it is still low. The best performing model, a ViT, achieves only 37% Top-1 accuracy (See Figure 2).

This work confirms prior results [7] such as ImageNet-A and demonstrates that today's neural classifiers have difficulty recognizing "natural adversarial examples" of known classes with different distributions. In addition, it points to the promise of augmented reality egocentric video data as a potential plentiful source of novel training data.

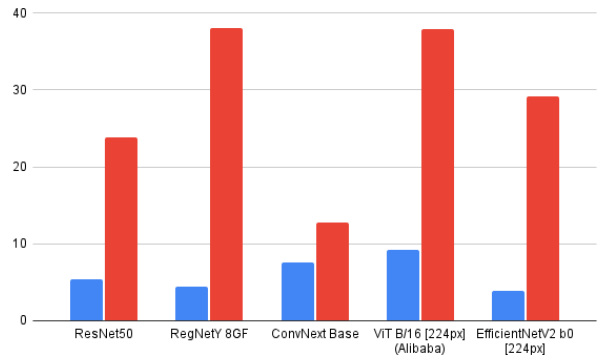

Figure 2. Original zero shot Top-1 accuracy on the Ego4D Subset compared with finetuned performance on 1,626 examples. Most models except ConvNext Base improved significantly, but even the best Top-1 accuracy is still relatively low at 37%.

B. Related Work

The robustness of neural classification models, their inability to generalize to unseen examples, and their tendency to memorize spurious correlations has been increasingly analyzed in recent work. Even a new similarly curated dataset to the very popular ImageNet demonstrated that with only slightly more difficult examples, neural classifiers lose 11-14% accuracy [14]. As introduced above, models have been found to engage in "shortcut learning" [3], where they rely on specious correlations found in image training datasets but not in the real world. Consider a green hillside, which is devoid of animals, but a neural model may predict "sheep grazing" due to the normal occurrence of sheep on such backgrounds. These models can misclassify a cow on a beach, because cows are normally not found there! [3] Deep Neural Networks (DNNs) generalize but very differently than humans. Further work investigated models' reliance on texture and demonstrated via a simple test showing images with conflicting texture versus shape that neural classification models do not generalize based on shape, but largely on texture [4]. When given the same test, humans demonstrated learning based on shape. Over-reliance on spurious correlations leads neural image classifiers to perform poorly in more complex real-world scenarios. The authors suggest in [3] that a good out-of-domain (OOD) generalization test should include: (1) a clear distribution shift, (2) a well-defined intended solution, and (3) a test where the majority of current models struggle. The new Ego4D egocentric data fulfills these criteria, which I will discuss further below.

Related to these efforts to investigate the lack of real-world understanding in learned representations are natural adversarial examples, which induce common models to make incorrect predictions. The ImageNet-A and ImageNet-O datasets [7] contain images filtered by hand

which cause common models to generate incorrect predictions with high confidence. ImageNet-A has classes common to ImageNet, while ImageNet-O contains classes outside ImageNet but which common models are not able to classify as unknown. Prior to this work, adversarial testing had largely involved "artificial adversarial examples, which are examples perturbed by nearly worst-case distortions" [7], while ImageNet-A and ImageNet-O use only natural examples. Evaluating on a range of models, the authors find that most do very poorly and can score as little as 2% accuracy. However, they do find that increasing model size and using some architectural modules such as self-attention can slightly improve performance. One downside to this work is that it required a large amount of human filtering. So much so, that graduate students were asked to memorize the 1,000 ImageNet classes in order to avoid images that contained more than one class. On the other hand, a higher percentage of egocentric data may be more naturally challenging due to its highly different distribution.

Other work [9] further investigates failure modes uncovered by ImageNet-A and finds that model performance suffers on that dataset due to three issues: multiple objects in a single image, object classes against unusual backgrounds, and small size relative to that found in ImageNet. To address the second and third issues, [9] releases a new dataset where they more closely crop the ImageNet-A images, and they see improved performance as a result. For the first, in spite of the intensive human-based filtering described above, the ImageNet-A dataset contains a significant subset of images with multiple objects. This is important to note for the construction of the Ego4D Subset described in this work.

These model weaknesses in terms of robustness and out-of-domain generalization suggest that there is still significant work to do in this area, that natural examples may be very valuable, and that having a clear distributional shift is important. As referenced above, the egocentric video data in the newly released Ego4D [5] dataset fits these criteria with a diverse set of natural examples from an egocentric perspective uncommon to previous computer vision datasets. Egocentric video looks wildly different than static image datasets and than other video datasets. The perspective, the lack of viewing angle stability, and other aspects introduce distinct complexities. The Ego4D dataset is wide ranging and was collected by a consortium of universities from around the world. Currently, the dataset contains 3,670 hours of video from camera-wearing volunteers, which span hundreds of settings and activities (household, outdoor, workplace, leisure, etc.) The video data was also collected in a range of geographies across 74 locations in 9 countries.

The Ego4D dataset contains 5 major video benchmarks, described further in the Data section. Within each, the dataset team has released data and annotations for specific subtasks as well as model baselines. As introduced above, I use data from the Hands & Object benchmark, which contains video footage of participants manipulating objects with their hands. Interestingly, there is some evidence already that existing models to not generalize well to this egocentric data. The most challenging of the Hands & Objects subtasks is a frame-wise binary classification task to identify object state changes (such as burning, splitting, etc) in a video. Existing models achieve very low scores on it with a maximum of 15 average precision.

In this work, I evaluate several recent popular neural classification models of various sizes and designs. I describe the high-level architectures briefly given the reader's likely familiarity with these models and leave discussion of the specific hyperparameters used to the Methods section. The first model evaluated is the classic convolutional model architecture composed of residual blocks, the ResNet [6]. This architecture contains residual blocks with connections that side-step the main block, which allow the network to learn the residual from the prior layer rather than the entire input.

A more recent model variant using residual blocks is the RegNetX architecture. This was developed as a result of a new "design space" of potential model architecture families, with the constraint that the models' blocks have varying widths according to a linear function. The models common design is composed of a stem, body, and head. The body has stages, and the stages have residual blocks. RegNetY is the RegNetX architecture plus the Squeeze-and-Excitation block [8], and it achieves superior performance to RegNetX. The authors state that RegNets are five times faster to train than the EfficientNet [18] on GPU, and they compared hundreds of different architectures in their study.

After success in NLP, the VisionTransformer (ViT) [2] was introduced as a model using only self-attention without any convolutional layers. It models images as a sequence of patches and achieved comparable performance to convolutional networks when pretrained on a large dataset. I use the AugReg [17] released models. That work studied the interplay between data augmentation, regularization and performance. The authors released 50,000 pretrained models. They found models that with increased compute could match those trained on much larger, non-public datasets.

After VisionTransformer, ConvNext [10] was developed as a purely convolutional architecture that claims to outperform ViTs on several tasks including COCO detection while requiring less compute. It has similar blocks to ResNet, but instead of BatchNorm uses LayerNorm and instead of ReLU uses GeLU. Separately, the EfficientNet work [18] introduced a way to scale the depth, width and image resolution systematically using a single compound coefficient, which is not dissimilar to the RegNet approach. The intu-

Figure 3. Examples from the Ego4D dataset that demonstrate its difficulty. Upper left: the mop is seen from an unusual perspective. Upper right: objects may be small relative to the size of the image. Bottom left: images may be cluttered with many objects. Bottom right: unusual tools and objects are common. Both the bottom conditions are filtered out of the new Ego4D Subset used in this work.

ition is that network dimensions should increase as image size increases. The authors also released a set of models based on a baseline found using neural architecture search. Later, EfficientNetV2 [19] followed on the initial work with even more efficient models improved by new specialized operations. In addition, they progressively train on larger images, and though this technique alone can decrease accuracy, they balance this with adaptive regularization through dropout and data augmentation. They argue that the V2 architecture outperforms the ViT by 2.0% for Top-1 accuracy on ImageNet-1K while training 5-11X faster.

C. Data

As introduced above, all evaluations in this work are performed on a subset of the recent Ego4D dataset [5] called the Hands & Objects Benchmark. Ego4D contains 3,670 hours of egocentric video data as of the present date, and the data comes from almost 1,000 camera wearers across 74 worldwide locations. The dataset is split into full video and shorter clips with annotations around particular events. Ego4D contains 5 major benchmarks with annotations specific to each. The benchmarks are: Episodic Memory (visual and language queries to video such as "where did I leave my keys?"), Hands and Objects (videos of participants manipulating objects with their hands), Forecasting, Audio-Visual Diarization (localize the speaker), and Social (predict if a speaker was speaking to the camera-wearer among other tasks in a multi-person setting).

The Hands & Object data contains 88,585 video clips total across training, validation, and test. In the training set, there are 41,085 clips total split approximately evenly between positive (has a state change) and negative (does not

Ego4D Paper Statistics	
Hours of Data	196.2
Number of clips	88,585
Average length	8.0 sec
"Change Objects in Train Split"	19,347
Processed Dataset Statistics	
Total Clips Listed in Dataset Index	19,071
Duplicate Clip + Frame	107
Clip Had No Object Label	120
Total Parsed	17,754

Figure 4. Top table: Ego4D Paper Statistics for Hands & Objects. Bottom table: actual data downloaded and processed. The total clips listed in the index file that accompanies the dataset were slightly inconsistent. In addition, while individual frames contained multiple annotations, a small number of images were also duplicated; these were eliminated during processing.

ImageNet 1K vs 21K	IN-1K	IN-21K
Total Images	17,754	17,754
Images with $\geq$1 IN class	3,474	8,488 [4,835]*
Images with $>$ 1 IN class	153	1,090
Invalid images	1	1
Images with 1 IN class	**3,321**	**7,398**
Total IN Classes	1000	21,483
Ego4D IN Classes	69	248 [419]*

Figure 5. Statistics of the processed Ego4D Subset when mapped to ImageNet-1K and ImageNet-21K classes [1]. *4,835 images have multiple distinct ImageNet-21K classes in a single annotation, which when included leads to 419 distinct classes in the Subset. If only the first ImageNet-21K class is included, there are 248 unique classes. Consequently, ImageNet-21K classification was phrased as an "any of" multilabel classification problem.

have a state change) examples. According to the paper, there are 19,347 "object of change" annotations, though I found slightly fewer when parsing the dataset's accompany annotations index file. The video prediction tasks for this benchmark involve: (1) localizing temporally the "point of no return" for a state change, (2) identifying a state-changing object before, during, and after its state change, and (3) classifying if an object change has taken place in a given frame. Examples of state changes of objects in the hand are burning, splitting, etc. While I do not attempt the video prediction tasks, the dataset includes annotated static image frames in order to identify the object of change. For each object state change, there is a "Pre" image before the change, a"PNR" (point of no return) image, and a "Post" image. I filter for the"Pre" images, because the state change event could involve the destruction of the object or a significant change in its appearance.

The filtering of the dataset and mapping the annotated

classes to ImageNet-1K or 21K [1] classes sets required several types of filtering and processing. Each clip has multiple frames annotated; each frame has multiple objects annotated; and each annotation has multiple words or lemmas associated with it. Annotations of hands had to be filtered out. A single Ego4D annotation may contain duplicated words or lemmas in many cases. A small set of images also seemed to be annotated twice, and those were removed. Some images contained multiple ImageNet classes across different annotations, and therefore those images were filtered out. See Figure 5 for statistics and the Appendix H for a list of unique ImageNet-1K classes found in the data.

For the ImageNet-21K experiments, there was a significant issue where over half the images (4,835 out of 8,488) with an ImageNet class actually contain a single annotation which has multiple distinct ImageNet-21K classes. This is an example annotation: "cloth(cloth,fabric,garment,kanga,rag)". The synonyms in parentheses describe a single annotation but often map to distinct ImageNet-21K categories. (Note: this is in contrast to the prior statement above where there are multiple ImageNet classes across separate annotations for the same image.) This seems to a problem related to the breadth of the classes, which is much larger than ImageNet-1K, and how the annotation object categories were obtained. It is related to the known problem, where objects may be labeled inconsistently across different examples with some labeled as a narrower category (e.g. "chair") and others labeled more generally (e.g. "furniture") [15]. In [15], this issue is addressed by using a semantic tree to relate parent (e.g. "furniture") and child (e.g. "chair") categories.

D. Methods

Each of the popular neural classification models introduced above is evaluated zero shot on the new Ego4D Subset. Below is a list with more information on architectures and network design. All models are built with PyTorch [12] or wrapped with it and come from one of two open source repositories: pytorch-image-models [20] or torchvision. I also created custom pipelines for video data filtering, image data loading, class mapping, zero shot evaluations, and finetuning. These were based on the two open source repositories above, but each step required significant implementation and refactoring. For visualization using gradient-weight class activation maps (GradCAM), I started with an open source implementation, but ultimately re-implemented it for my needs.

Image input sizes are transformed to 224x224x3 pixels unless otherwise indicated. Other transformations vary by model but are standard for each and used as per the pytorch-image-models implementation. Most models include the transformation to normalize by the BatchNorm mean and standard deviation from the ImageNet dataset for the three channels. Most models also transform the input through cropping. Models used were limited to what was available through the repositories described above in order to limit the amount of integration code required. Model parameter sizes are listed alongside the results in Figure 7.

- ResNet50, 101, 152: These are ResNet models with 50, 101, or 152 layers.
- RegNetY 8GF, 32GF: GF refers to GigaFLOPs and is the number of FLOPS used to train the model. 32GF is the largest released model. [1]
- VisionTransformer (ViT): I evaluate both Base (12 Layers) and Large model (24 Layers) sizes with 16x16 pixel patch size. I use the AugReg [17] models. The main models released that are usable for zero shot evaluation were pretrained on ImageNet 21K and finetuned on ImageNet 1K.[2]
- ConvNext: I evaluate both the Base model with number of channels per stage equal to (128, 256, 512, 1024) and blocks per stage (3, 3, 27, 3) and the Large model, which has channels (192, 384, 768, 1536) and blocks (3, 3, 27, 3).
- EfficientNetV2: I evaluate 5 EfficientNetV2 models of varying size and pretraining: b0, b3, and Small, Medium, Large. The b0 model is the smallest with only 5.3M parameters and 224x224 pixel inputs. The b3 model has 14M parameters and takes 300x300 pixel inputs. The other three larger models (S/M/L) take increasingly larger input with Medium and Large at 480x480 pixels.

As noted above, [9] found that ImageNet-A classes were often misclassified due to their small size versus the background relative to the class' average size in the ImageNet dataset. I cannot repeat their exact experiment which involved cropping the images more closely to the object, but I do evaluate different ViT models with the same architecture except for increased input size as a proxy. In addition, where possible, I compared ImageNet-1K trained models with models of the same architecture that were trained on ImageNet-21K and finetuned on ImageNet-1K. Not all model combinations are available.

I also evaluate two models, ConvNext Large and EfficientV2 Large, that were pretrained solely on ImageNet-21K. I would have liked to evaluate one of the ViT Large's that Google released, but they zeroed out the classification heads for the ImageNet-21K models, which therefore can only be used for finetuning not zero shot evaluation.

[1] For other training details and hyperparameters, the interested reader can look at the original authors' model zoo.

[2] See more information about the AugReg released models here.

Versus the 1,000 classes in ImageNet-1K, the 21K dataset contains many more classes (21,843). As introduced in the Data section, a single Ego4D annotation, which other than the main class lemma should contain synonyms for an object, often had multiple distinct ImageNet- 21K classes. This was the case in over half the images. Consequently, in this work I perform ImageNet-21K evaluation as an "any of" multilabel classification problem: if the model's top-k predictions contain any of the labels present from a single annotation with synonyms, the example is judged as correct. Multilabel subset accuracy, which would require that all labels present be correctly predicted, is not used.

Finally, I also finetune the smallest ImageNet-1K models in each of the five model families to investigate the models' ability to learn the egocentric data. While finetuning all the models would have been most informative, the smallest models - ResNet50, ConvNext Base, ViT Base, and EfficientNetV2 b0 - were more suitable due to limited compute. I split the Ego4D Subset roughly in half, with 1,626 images in training and 1,694 in validation. Many of the static images come from the same clips, and each video also contains several clips. To address this problem, I split the data in half by clip, not by image, for training and validation. Before doing the splitting this way, I found that there was too much similarity in the train and validation sets due to the nature of the Ego4D data: each video involves a single activity and setting. This allowed the models to erroneously overfit the training set without penalty on the validation set. For example, ConvNext Base achieved 72% Top-1 validation accuracy with the former splitting regime but only 12.81% with the clip splitting strategy.

The models were trained with hyperparameters of: SGD with Momentum of 0.9; 3 warm-up epochs with warm-up learning rate 1e-4 and 5e-2 after that; learning rate decay was 0.1 per epoch. Batch size was 128. Models were trained on a single V100 or A100 GPU through Google Colab. The standard transformations were applied per model. The ImageNet BatchNorm mean and standard deviation transformations were applied to all five. Other transformations were: 0.5 probability of horizontal flip, and color jitter with probability 0.4 factor. The models were trained for a maximum of 50 epochs with 5 epochs of validation patience before terminating.

I evaluate all models on Top-1 and Top-5 accuracy for both zero shot and finetuning evaluations. Finetuned evaluations are reported on the validation set, which was also used for early stopping during finetuning, but this was unavoidable due to the small amount of data in the Ego4d Subset. I also perform quantitative analysis looking at high confidence erroneous predictions and qualitative analysis using the GradCAM [16] visualizations.

Figure 6. High confidence ($p > 0.90$) but incorrect predictions by ResNet50 on zero shot evaluation. Top Left: label was "vacuum", but model predicted "seat belt", which is also in the image. Top Right: model predicted oxygen mask but label was "wool". It is unclear what exactly that scene is. Bottom left: model predicted "pot" but label was "vase". Bottom right: model predicted "cleaver" but label was "wallet".

E. Results & Analysis

All models tested perform poorly on this new Ego4D Subset. The highest zero shot Top-1 accuracy is achieved by ConvNext Large pretrained on ImageNet-21K with finetuning ImageNet-1K and is only 18.7% (See Figure 11). For ImageNet-1K pretrained models only, EfficientNetV2 Large achieved the best Top-1 accuracy score at approximately 17% (See Figure 7). In that setting, EfficientNetV2 did significantly better than the other convolutional models. In addition, ResNet, RegNetY and ConvNext did not seem to benefit from increased model size. However, EfficientNetV2 and ViT did. Pretraining on ImageNet-21K and finetuning on ImageNet-1K did provide a boost on all models tested. Using the same model architecture and increasing image input resolution also improves accuracy, at least for the ViT I was able to test (See Figure 8).

Moreover, models also produced high confidence predictions that were incorrect when evaluated zero shot. ResNet50 did that on 3.6% of all examples with a high threshold of 0.90. See Figure 6 for examples of erroneous high confidence predictions. Interestingly, one failure mode still seems to be having multiple objects in the image, in spite of my aggressive filtering (top left image: "seatbelt" prediction versus "vacuum" label). Other failure modes are having a scene that is hard to decipher even for a human (top right), and slightly different labels (bottom left: prediction "pot" vs label of "vase"). The bottom right example where the model predicted "cleaver" and the label was "wallet"

ImageNet 1K-Trained Models on Ego4d Subset	Top-1	Top-5	Params (M)
ResNet50	5.42	12.02	25.6M
ResNet101	6.08	14.13	44.5M
ResNet152	6.17	13.77	60.2M
RegNetY 8GF	4.37	10.87	11.2M
RegNetY 32GF [288px]	7.98	17.59	19.4M
ConvNext Base	7.62	16.02	88.6M
ConvNext Large	7.89	17.56	197.8M
ViT B/16 [224px] (Alibaba)	9.28	17.44	86.5M
ViT B/16 [224px] (Google)*	12.41	24.4	86.5M
ViT L/16 [224px] (Google)*	15.87	**30.03**	304.3M
EfficientNetV2 b0 [224px]	3.95	10.75	7.1M
EfficientNetV2 b3 [300px]	10.31	20.87	14.4M
EfficientNetV2 S [384px]	10.3	20.72	21.5M
EfficientNetV2 M [480px]	11.75	20.81	54.1M
EfficientNetV2 L [480px]	**16.96**	27.98	118.5M

Figure 7. Zero shot Top-1 and Top-5 accuracy for models trained on ImageNet 1K. *Google ViTs were trained on ImageNet 21K and finetuned on ImageNet 1K. All models do poorly on this new dataset composed from Ego4D. However, larger models do have the best, albeit low, performance. The best Top-1 score is $\approx 17\%$.

Vary Models Image Resolution on Ego4d Subset	Top-1	Top-5	Params (M)
ViT B/16 [224px] (Google)*	12.41	24.4	86.5M
ViT B/16 [384px] (Google)*	15.45	27.95	86.9M

Figure 8. Larger image input size (224px versus 384px) slightly improves accuracy for models of otherwise same architecture.

ImageNet-21K Trained	Top-1	Top-5	Params (M)
ConvNext Large [224px]	1.31	5.27	229.8M
EfficientNetV2 L [480px]	2.41	7	145.2M

Figure 9. These two models were trained only on ImageNet 21K, and the Ego4D Subset annotations were mapped to one of 21,483 classes. Classification was performed as an "any of" problem due to multiple distinct ImageNet 21K classes present in a single Ego4D annotation.

seems to be a genuine misclassification.

Unfortunately, the ImageNet 21K pretraining only model accuracies are low (See Figure 9). I carefully implemented the multilabel-like "any of" evaluation method, including writing tests for the metric, but I think with better mapping to ImageNet-21K classes, performance could possibly be increased. However, this low performance could also be explained by the combination of factors affecting the ImageNet-1K evaluations compounded by the multiple distinct classes within individual Ego4D annotations and the inconsistent labeling in ImageNet-21K [9].

Pre-training	ImageNet-1K vs ImageNet-21K Finetuned 1K	Top-1	Top-5
IN-1K	ViT B/16 [224px] (Google)*	9.28	17.44
IN-1K	ViT L/16 [224px] (Google)*	N/A	N/A
IN-1K	EfficientNetV2 L [480px]	**16.96**	**27.98**
IN-1K	ConvNext Large	7.89	17.56
21K-1K	ViT B/16 [224px] (Google)*	12.41	24.4
21K-1K	ViT L/16 [224px] (Google)*	15.87	30.03
21K-1K	EfficientNetV2 L [480px]	18.31	32.53
21K-1K	ConvNext Large	**18.73**	**32.68**

Figure 10. Top models are pretrained only on ImageNet-1K whereas bottom models are trained on ImageNet-21K and then finetuned on ImageNet 1K. You can see modest improvements across the board. ConvNext Large performance on ImageNet-1K versus the finetuned version has a larger jump than other models.

	Finetuned on Ego4D Subset	Top-1	Top-5
Original	ResNet50	5.42	12.02
Original	RegNetY 8GF	4.37	10.87
Original	ConvNext Base	7.62	16.02
Original	ViT B/16 [224px] (Alibaba)	**9.28**	**17.44**
Original	EfficientNetV2 b0 [224px]	3.95	10.75
Finetuned	ResNet50	23.79	51.06
Finetuned	RegNetY 8GF	35.66	**63.28**
Finetuned	ConvNext Base	12.81	22.73
Finetuned	ViT B/16 [224px] (Alibaba)	**37.96**	62.93
Finetuned	EfficientNetV2 b0 [224px]	29.16	56.67

Figure 11. Model performance after finetuning on 1,626 examples of the Ego4D Subset. The ViT Base performance improves the most, while the ConvNext Base model seems not to learn as well from the new data.

The finetuning results demonstrate that most of the models improve significantly when shown examples of the egocentric distribution (See Figure 2). The RegNetY and ViT both achieve almost 38% Top-1 Accuracy on the held out validation subset from well under 10% zero shot. However, overall that performance is still quite low. ConvNext Base performance is an anomaly and did not increase much. The ConvNext Large performance also jumped the most in the ImageNet-1K pretraining versus IN-21K finetuned on ImageNet-1K. Therefore, it seems possible that the ConvNext models either are not implemented properly in pretrained versions I used or have unique weaknesses.

F. Conclusion

In this work, I filter egocentric data for static images with objects common to ImageNet-1K and ImageNet-21K. I evaluate common neural classification models of varying sizes from five different model families, both convolutional and Vision Transformers. Zero shot performance is very

low across models on this new Ego4D Subset. ImageNet-21K finetuned on ImageNet-1K models perform slightly better as do models which large size or which take higher resolution image inputs. However, only finetuning demonstrates significant gains in accuracy, though the highest Top-1 accuracy achieved by the ViT Base model of 38% is still relatively low. This work demonstrates that egocentric data are adversarial examples for today's classifiers and degrade their performance significantly.

This project has its limitations, and there are several opportunities for future work. One opportunity is to do better image filtering of the original Ego4D data, potentially by humans. Cropping images with multiple object categories as in [9] would also allow more images to stay in the Subset. In addition, it would obviously be great to evaluate more models, larger models, and do more finetuning. Some of the comparisons I was forced to make were imperfect: for example, I had no ViT Large model trained only on ImageNet-1K (versus ImageNet-21K and finetuned on ImageNet-1K). Data augmentation techniques such as those evaluated in [7] and [17] would be interesting to explore. For ImageNet-A, data augmentation did not help performance. I would also have liked to do more error analysis as to why the ImageNet-21K performance is so low.

G. Contributions & Acknowledgements

I completed this project by myself. I had no external collaborators, nor did I make use of external compute. I did use pretrained models and starter code from the repository pytorch-image-models which has some pretrained classification models. I created the custom pipeline to extract the right video frames from the dataset, process them to match ImageNet classes, created a custom Pytorch dataloader, custom evaluation logic both for single label and multilabel cases, and custom finetuning scripts. I also investigated the use of Detectron2 to evaluate object detection models, but that work did not end up in the final paper.

H. Appendix

Below is a list of the 69 ImageNet-1K classes found in the dataset. They are concentrated in the home or in an industrial setting given the distributed of data from the camera wearers. This list could possibly be expanded by revising the filtering criteria, but many of the images contain multiple objects, which is difficult to overcome.

{'rubber', 'broom', 'notebook', 'speaker', 'grocery', 'chainsaw', 'nail', 'iron', 'cardigan', 'screwdriver', 'file', 'envelope', 'radio', 'hook', 'wallet', 'dough', 'wool', 'bucket', 'pot', 'apron', 'dishwasher', 'mask', 'bookcase', 'remote', 'dustbin', 'television', 'strainer', 'sock', 'spatula', 'pole', 'trolley', 'packet', 'scale', 'sunglass', 'torch', 'shovel', 'mower', 'screen', 'plane', 'switch', 'vase', 'cellphone', 'stove', 'basketball', 'hammer', 'mushroom', 'cup', 'board', 'blower', 'plate', 'sandal', 'napkin', 'mortar', 'banana', 'mouse', 'refrigerator', 'dumbbell', 'mop', 'jean', 'pillow', 'pizza', 'cucumber', 'screw', 'vacuum', 'tray', 'paintbrush', 'snorkel', 'frypan'}

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1, 2, 4, 5

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 2, 3

[3] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *ArXiv*, abs/2004.07780, 2020. 1, 2

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ArXiv*, abs/1811.12231, 2019. 1, 2

[5] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Q. Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, F. Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. *ArXiv*, abs/2110.07058, 2021. 1, 3, 4

[6] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2, 3

[7] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. Natural adversarial examples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15257–15266, 2021. 1, 2, 3, 8

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 3

[9] Xiao Li, Jianmin Li, Ting Dai, Jie Shi, Jun Zhu, and Xiaolin Hu. Rethinking natural adversarial examples for classification models. *ArXiv*, abs/2102.11731, 2021. 3, 5, 7, 8

[10] Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. 2022. 3

[11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. 01 2022. 2

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[13] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10425–10433, 2020. 2

[14] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *ArXiv*, abs/1902.10811, 2019. 1, 2

[15] T. Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *ArXiv*, abs/2104.10972, 2021. 5

[16] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2019. 1, 6

[17] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *ArXiv*, abs/2106.10270, 2021. 3, 5, 8

[18] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019. 2, 3

[19] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. *ArXiv*, abs/2104.00298, 2021. 2, 4

[20] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5