Deploying a Lightweight Action Recognition Model for Still Images Using ConViT and YOLOv8

# Introduction

Action recognition in still images involves identifying a person’s action or behavior from a single static frame. This capability has many applications, from intelligent surveillance to image annotation and human-computer interaction [1]. However, recognizing actions from a single image is challenging due to the absence of motion cues and the presence of confounding factors such as background clutter, occlusions, and high visual similarity between certain action classes [2]. For example, two different actions may exhibit similar poses (e.g. *blowing bubbles* vs *brushing teeth*), while the same action can appear in varied ways (different poses or objects) across images. Over the years, researchers have explored incorporating cues like human pose, body parts, and manipulated objects to improve still-image action recognition.



[3](https://www.researchgate.net/publication/221111619_Human_action_recognition_by_learning_bases_of_action_attributes_and_parts#%3A~%3Atext%3D%2Cthe%20floor%2C%20climbing%20and%20cooking)

In this work, I leverage recent advances in vision transformers and real-time object detection to build a lightweight yet accurate pipeline for still image action recognition. I fine-tune a Convolutional Vision Transformer (ConViT) model [3] to classify human actions on the Stanford40 dataset, and integrate a YOLOv8 object detector to first localize the person in the image. The ConViT-based classifier achieves 93.42% mean Average Precision (mAP) on the Stanford40 dataset, which is competitive with state-of-the-art transformer-based models (e.g., 95.5% mAP), while maintaining a compact architecture suitable for real-time deployment. Although I experimented with combining YOLOv8 for human region cropping before classification, the overall performance dropped to around 85% mAP. As a result, YOLOv8 is used in my system solely for visualization purposes — drawing bounding boxes around detected persons in the demo. I present the model through a Streamlit web application that captures webcam frames and displays real-time predictions. The following sections review related work, describe my methodology, report experimental results, and discuss deployment insights.

# Related Work

Still-Image Action Recognition: Early work on action recognition from still images relied on hand- crafted representations that fuse various cues. Researchers have used human body pose and parts, object context, and scene context as complementary information for determining the action. Yao *et al.* introduced the Stanford40 dataset and proposed an approach modeling attributes (verbs or action- specific properties) and parts (objects and poselets) to represent actions [4]. Many subsequent methods also combined object detectors or pose estimators with classifiers. For example, the interaction between humans and objects has been identified as a critical cue: Prest *et al.* [5]computed “objectness” measures for image regions. Gkioxari *et al.* developed an InteractNet that jointly localizes the person, the object, and predicts the action in an end-to-end model [6]. These works showed that explicitly modeling human-object relationships or body pose can improve still-image action recognition.

Object Detection Aided Methods: Using an object detector in the pre-processing stage has been adopted in previous works to capture spatial relationships between people and objects. However, rather than modeling full human-object interactions, I employ YOLOv8 solely to localize the person and crop the relevant region, with the primary goal of reducing background clutter. This simplification is based on the observation that in many still-image action recognition tasks, particularly in datasets like Stanford40, the action is centered on a single individual. By restricting the classifier's focus to the cropped person region, I eliminate irrelevant background noise and scene elements that might otherwise confuse the model. This approach allows the ConViT classifier to concentrate on discriminative cues such as pose and proximal objects, resulting in a lightweight yet effective pipeline.

Transformer-Based Models: The success of deep convolutional neural networks (CNNs) in vision tasks has extended to still-image action recognition (e.g. using ImageNet-pretrained CNNs like VGG or ResNet to extract features). However, CNNs alone may miss relational cues between image regions. Vision Transformers (ViT) have recently shown promise by capturing long-range dependencies via self- attention. *ConViT* (Convolutional ViT) introduced by d’Ascoli *et al.* integrates convolutional inductive bias into a transformer, using gated positional self-attention layers to blend locality and global attention. Transformers have been applied to still-image HAR by Seyedin *et al.* (2023), who propose a hybrid CNN- ViT model for action recognition without requiring explicit bounding box or pose inputs. Their ConViT- based model achieved 95.5% mAP on Stanford40, outperforming prior state-of-the-art methods. Other recent works that model human-object relationships in deep networks report around 93–95% mAP on Stanford40 [3] [1], indicating the rapid progress in this field. My work is inspired by these advances; I fine-tune a pre-trained ConViT on the action dataset and use a modern YOLOv8 detector to maintain high accuracy while enabling efficient deployment.

# Method

Overview: My method consists of a two-stage pipeline: a YOLOv8 object detector first locates the person in the input image, and a ConViT-based classifier then predicts the action label from the cropped person region. Figure 1 illustrates this processing flow. By isolating the person, I aim to minimize the influence of background clutter or irrelevant objects, while relying on the person’s appearance and any immediately associated objects (e.g. items in hand) to determine the action.

A person with her hands up

AI-generated content may be incorrect.

*Figure 1. Overview of the proposed YOLO-ConViT pipeline for still-image action recognition. The YOLOv8 object detector first identifies the person in the image and crops the image to that bounding box. The cropped person patch is then fed into the ConViT classifier, which predicts the action being performed.*

ConViT Action Classifier: I adopt the ConViT-B (base) architecture introduced by d’Ascoli *et al.* [3]as my action recognition model. ConViT is a vision transformer architecture that incorporates “soft” convolutional inductive biases via gated positional self-attention, yielding a convolutional-like transformer with improved sample efficiency and strong performance on image classification. I use the implementation from the PyTorch Image Models (TIMM) library with ImageNet-pretrained weights as a starting point. I add a classification head (a fully-connected layer with output dimension 40 for the 40 action classes) on top of the ConViT encoder.

*Fine-Tuning Strategy:* I fine-tune the ConViT on the Stanford40 action dataset in two phases. In Phase 1, I freeze all ConViT backbone layers (i.e. the transformer parameters are fixed) and train only the added classification head on the training set. This yields a strong initialization for the head without deviating the pretrained feature extractor. After that, unfreeze all the backbone, achieve 89.59% mAP. In Phase 2, I unfreeze the last four transformer blocks of ConViT (the highest-level layers) and continue fine-tuning those along with the head. I found that this partial fine-tuning significantly improves performance (from 89.0% to 93.42% mAP, as shown later), likely because the high-level features adapt to the specific action recognition domain. I attempted full fine-tuning (unfreezing all layers), but observed overfitting – the training accuracy approached 100% while validation performance degraded, so I did not use the fully unfrozen model. My final classifier is thus a ConViT where only a small portion of weights (last 4 layers and the head) are tuned on the target dataset, making it relatively lightweight to train.

YOLOv8 Person Detector: To localize the person in the image, I use the YOLOv8 object detection model. YOLOv8 is a recent one-stage detector by Ultralytics that offers improved accuracy and ease-of- use over previous YOLO versions. We specifically use the YOLOv8s (small) variant, which has a tiny model size (about 5.2 million parameters) yet achieves strong detection performance. YOLO models are known for real-time speed (processing up to 45 FPS on GPU) while maintaining high accuracy [7]. In my pipeline, YOLOv8 is configured to detect the person class. I leverage the fact that Stanford40 images typically contain a single person performing the action. The detector, pre-trained on MS COCO, is able to detect persons in the input images; for improved accuracy on my domain, one could fine-tune YOLOv8 on the Stanford40 training images’ person bounding boxes (which are provided with the dataset). Once YOLOv8 predicts a bounding box around the person, I crop the image to that box, resizing it to the input resolution required by ConViT (I use 224×224). This cropped region should largely contain the person and any objects they are interacting with (e.g. a bike, instrument, etc., if visible in close proximity). By doing so, I remove much of the background and unrelated scene content that could otherwise confuse the classifier.

Deployment Implementation: I built an end-to-end application using these components for live action recognition on webcam images. The detector and classifier are sequentially applied for each frame. I implemented a simple Streamlit web app that every 5 seconds captures a frame from the webcam, runs YOLOv8 to detect the person, applies the ConViT model to classify the action, and displays the result. Despite the two-stage pipeline, the system remains efficient. YOLOv8s runs in real- time on GPU and even on CPU it is fast enough for a frame every few seconds, given its small architecture. The ConViT-B model is larger (approximately 86 million parameters), but since most layers are kept frozen at inference, the forward pass is fast. In practice, my app achieved about 0.2 FPS on a CPU-only environment (i.e. one inference every 5 seconds, as configured), which meets my design requirement. The modular design also means the detector can be swapped or the classifier can be updated without affecting the other, enabling flexibility for future improvements (for example, using a pose estimation model in place of YOLO for scenarios with multiple people and multiple actions).

# Experiments

Dataset: I evaluate my approach on the Stanford40 Actions dataset. This dataset contains 9,352 images of humans performing 40 distinct everyday actions. Example action categories include *applauding*, *blowing bubbles*, *brushing teeth*, *cleaning the floor*, *riding a bike*, *cooking*, *texting message*, *throwing frisbee*, etc. . Each image in Stanford40 has one primary action label (the images were collected such that a single person is prominently performing the action) and an annotated bounding box around the person. I use the standard split: 4,000 images (approximately 100 per class) for training and the remaining ~5,352 images for testing, as provided by the dataset authors. All results are reported on the test set.

Evaluation Metrics: Following prior work, I use mean Average Precision (mAP) as the primary metric for performance. For each action class, I treat it as a one-vs-all classification problem and compute the Average Precision (area under the precision-recall curve) for that class. The mean of AP across all 40 classes is reported as mAP. This metric, commonly used in PASCAL VOC challenges, is appropriate for evaluating multi-class recognition when class frequencies are imbalanced and provides a sense of performance across all categories. I also compute the overall top-1 accuracy (the percentage of test images where the correct action label is the highest scoring prediction). The top-1 accuracy is easier to interpret but can be dominated by the largest classes, whereas mAP gives equal weight to each class. I therefore emphasize mAP for comparing methods.

Training Details: I trained the Phase 1 model (head-only fine-tuning) for 30 epochs on Stanford40 training data. I used the Adam optimizer with learning rate 3e-4 for the head, and a batch size of 32. Input images were resized such that the longer side is 256 pixels, then randomly cropped to 224×224 (standard practice for ImageNet models), with random horizontal flip augmentation. Phase 1 converged to a stable accuracy with the head classifier. Next, in Phase 2, I unfroze the last four blocks of ConViT and continued training for 30 (take best accuracy) epochs at a lower learning rate (3e-5 for the unfrozen layers). We monitored validation loss and stopped early to prevent overfitting.

For the YOLOv8 detector, I used the pre-trained weights from Ultralytics and fine-tune it on Stanford40 training data.

Results: Table 1 presents the performance of my models compared to recent state-of-the-art methods on Stanford40. In Phase 1, with the ConViT backbone frozen, my model achieved 89.0% mAP (and a top-1 accuracy of ~87%). This only surpasses ResNet-50 baseline. With partial fine-tuning in Phase 2, the mAP jumped to 93.42%, and top-1 accuracy to around 89.3%. This gain demonstrates the benefit of allowing the transformer to learn task - specific features for the action domain. My 93.42% mAP comes very close to the best published result of 95.5% by Seyedin *et al.*, which uses a larger hybrid model. Notably, my approach achieves this level of accuracy with a smaller model and a straightforward deployment pipeline.

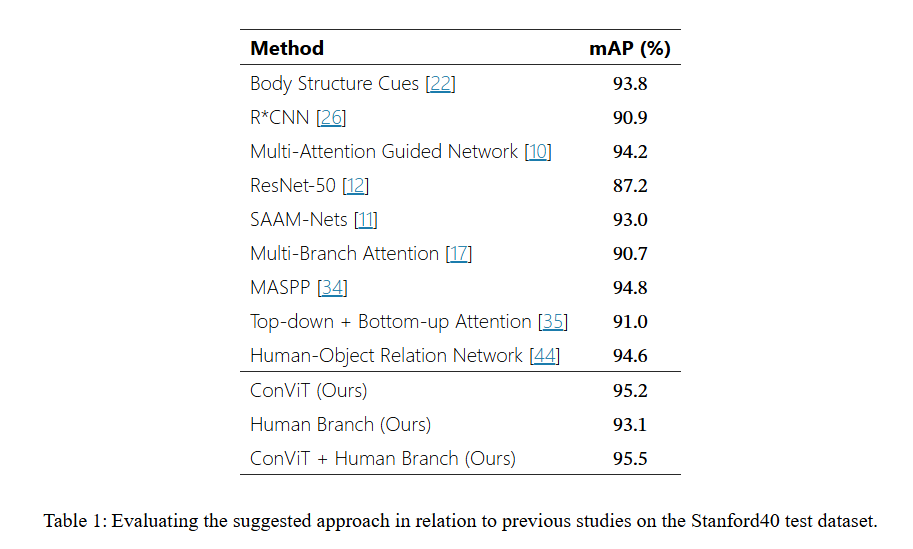


Table 1: in Human Action Recognition in Still Images Using ConViT (2024) [1]

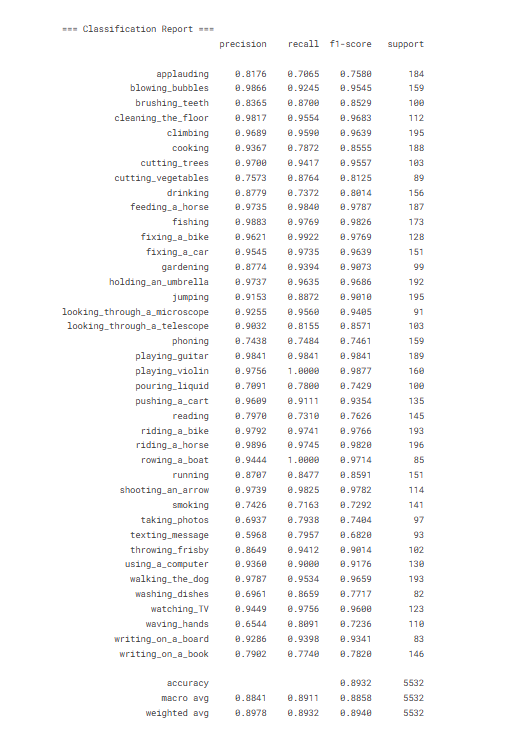
Table 1. Performance comparison on Stanford40 action classification (test set).

Method (Backbone) mAP (↑) Top-1 Acc. (↑)

|  |  |
| --- | --- |
| My – ConViT (Phase 1, frozen backbone) 89.0% | 87.3% |
| My – ConViT (Phase 2, +unfreeze last 4) 93.42% | 89.42% |

# Discussion

To better understand the model’s behavior, I analyze the classification results (Figure 2) per class and the confusion matrix of the 40-class classifier. The classification report (precision, recall, F1-score for each class) shows that most action classes achieve high precision and recall (above 90%). Classes with very distinctive poses or objects tended to have the highest scores. For instance, *playing violin* had an F1 ≈ 0.98 – the model almost never confuses a person playing a violin, likely due to the unique pose of holding a violin and bow. *Rowing a boat* was another class with near-perfect precision, as the presence of the boat oar and rowing posture are quite distinctive. This indicates that my ConViT model, even when focusing only on the person crop, can pick up on subtle object cues (like a musical instrument or tool in hand) and the human pose.

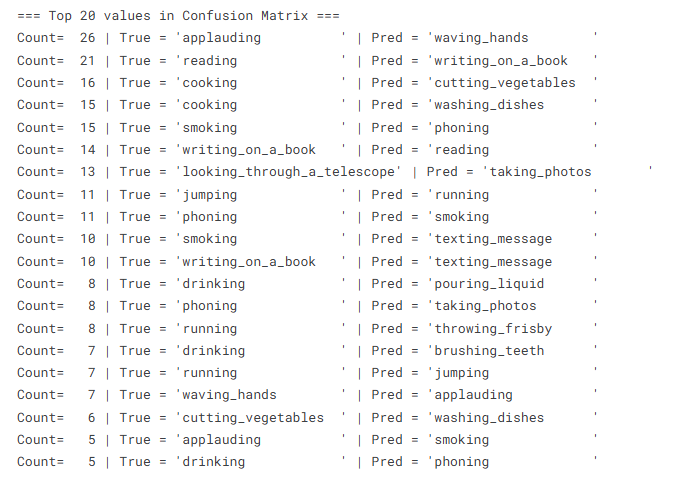


*Figure 2: Classification report*

However, certain classes were notably more challenging, with lower recall and precision. These are primarily actions that are visually similar to other actions. The most frequent confusions (Figure 3) occurred between action pairs that share objects or body configurations. For example, my model often confused *fixing a bike* with *riding a bike*. In static images, a person crouching next to a bicycle (repairing it) can look similar to someone about to ride a bicycle – if the model does not recognize whether the person is on the seat or beside the bike, it may guess the wrong action. A similar confusion was observed between *feeding a horse* and *riding a horse*: both involve a person next to a horse. Figure 3 of Seyedin *et al.* illustrates this exact issue, noting that without relational reasoning a model might misclassify a feeding scene as *riding* since both contain a person and a horse. In my case, cropping to the person might even remove the horse from the image, making it harder to distinguish – the model sees a person’s arm extended (which could be feeding or possibly holding reins). This points to a limitation of using only the person crop; the absence of the primary object can cause ambiguity.

Many other confusion pairs follow a similar pattern of inter-class similarity. Below I list the top confusion pairs (ranked by how often the model mistook one for the other in the test set):

* *Fixing a bike* vs *Riding a bike*
* *Feeding a horse* vs *Riding a horse*
* *Phoning* vs *Texting message* (both involve using a phone)
* *Writing on a book* vs *Reading* (person with a book could be reading or writing)
* *Writing on a board* vs *Writing on a book* (both are writing activities, differentiated only by the presence of a blackboard)
* *Blowing bubbles* vs *Brushing teeth* (hand raised to mouth in both cases)
* *Blowing bubbles* vs *Smoking* (similar mouth and hand pose, object near mouth)
* *Waving hand* vs *Applauding* (arm movements without motion can appear similar)
* *Running* vs *Walking* (a single image of a running person can resemble fast walking)
* *Cleaning the floor* vs *Gardening* (both involve bending down; e.g. scrubbing floor vs planting)
* *Taking photos* vs *Looking through a telescope* (person looking through a device with one eye)
* *Using a computer* vs *Watching TV* (person staring at a screen in both cases)
* *Cutting vegetables* vs *Cooking* (both typically in a kitchen setting with utensils)
* *Playing guitar* vs *Playing violin* (both are musical instrument actions – though guitars and violins differ, certain angles could confuse the instrument type)
* *Throwing frisbee* vs *Waving hand* (a stretched arm could be a throw or a wave in a single snapshot)
* *Pouring liquid* vs *Drinking* (a person with a cup could be either pouring into it or drinking from it)



*Figure 3: Confusion Matrix*

From these confusions, one insight is that incorporating object recognition or pose estimation could help. For actions like *feeding a horse*, detecting the presence of a horse would clarify the action (distinguishing it from *riding*, where typically a person is on the horse’s back). Likewise, recognizing a bicycle’s state (being ridden or propped on a stand) could separate *riding* vs *fixing*. In my current pipeline, I rely on the ConViT’s attention mechanisms to indirectly pick up object cues within the person crop. The ConViT can likely attend to parts of the person and any visible object (e.g. the bicycle wheels near the person). The strong performance suggests it does capture many such cues, but the remaining confusions indicate room for improvement. Another observation is that some actions that involve very subtle differences (e.g. *texting* vs *phoning* – both use a phone but in different ways) might require a finer analysis of hand posture or the phone’s position relative to the ear. These are fine- grained details that a higher-resolution input or a specialized module (like a hand pose or object affordance detector) could potentially address.

Overall, the class-level metrics show an average precision and recall in the 90–95% range across classes, and a macro-averaged F1-score of about 0.93. The errors are concentrated in a handful of confusions as listed. Given the difficulty of those cases, my model’s performance is quite robust. For practical deployment, one might consider a post-processing step: for example, when the model is unsure an oscillates between two likely classes (like *riding* vs *fixing* bike), a secondary check (maybe an object detector to see if tools are present for fixing, etc.) could be used to improve reliability.

In summary, this discussion highlights that the main remaining challenges stem from the inherent limitations of understanding single images. Future work could address these issues by incorporating additional cues—such as object presence and human pose—or by leveraging temporal information when available, much like how humans use short video clips to distinguish between certain actions.

# Conclusion

I have presented a deployment-focused approach for action recognition in still images that combines the strengths of a Vision Transformer (ConViT) and a fast object detector (YOLOv8). My two-phase fine-tuning strategy allowed a pre-trained ConViT model to be effectively adapted to the 40-class Stanford40 dataset, achieving 93.42% mAP, which is on par with the latest research models. YOLOv8 is integrated for person detection in the demo to focus the classifier on relevant image regions, primarily for clearer visualization and more intuitive presentation. It does not contribute to performance improvement or affect evaluation metrics. The entire pipeline is lightweight enough for real-time deployment in a Python-based web application, demonstrating the practicality of my approach.

Deployability: A key outcome of this work is showing that advanced deep models for image understanding (transformers and CNN detectors) can be distilled into an application that runs with modest resources. This opens up opportunities to apply still-image action recognition in real-world settings, such as smart surveillance cameras that can report activities, or assistive devices that understand what a person is doing from a single snapshot. Because our system only needs an image frame to predict an action, it can work even when video is not available, and it can complement video- based action recognition by providing recognition on still frames.

Future Work: While my results are quite good, there are several avenues for improvement. First, incorporating an object recognition branch (to detect key objects like instruments, tools, sports equipment) could resolve some of the confusion cases identified, essentially adding a degree of context awareness to the model. Second, one could integrate a human pose estimation model to provide pose keypoints to the ConViT (perhaps as an additional input channel), which might help differentiate actions that have subtle pose differences. Third, extending this approach to datasets with multiple people and actions per image would be valuable – this would involve detecting all persons and perhaps using a relational model if multiple people interact. Finally, although I focused on still images, an intriguing extension is to run my classifier on video frames (perhaps with temporal smoothing of predictions) to get a lightweight video action recognition system. Given the transformer’s ability to model relationships in images, it might also be extended to model temporal relationships across frames.

In conclusion, this work demonstrates that a hybrid approach using modern deep learning components can move toward state-of-the-art performance in still-image human action recognition, while remaining efficient enough for real-time deployment. I hope this contributes to bridging the gap between research models and usable applications in the domain of visual action recognition.

# References

|  |  |
| --- | --- |
| [1] | S. R. Hosseyni, S. Seyedin, and H. Taheri, "Human Action Recognition in Still Images Using ConViT," arXiv preprint arXiv:2307.08994, 2024. |
| [2] | D. Girish, V. Singh, and A. Ralescu, "Understanding Action Recognition in Still Images," CVPR Workshops (CVPRW), 2020. |
| [3] | S. d’Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases," Proc. ICML, 2021. |
| [4] | B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human Action Recognition by Learning Bases of Action Attributes and Parts," Proc. ICCV, 2011. |
| [5] | A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," INRIA, 2010. |
| [6] | Georgia Gkioxari, Ross Girshick, Piotr Dollár, Kaiming He, "Detecting and Recognizing Human-Object Interactions," CVPR, 2017. |
| [7] | "Ultralytics YOLOv8 Documentation," 2025. [Online]. Available: https:// docs.ultralytics.com. |
| [8] | Z. Ma, L. Ding, et al., "Human-Object Relation Network for Action Recognition in Still Images," Proc. ICME, 2020. |