Reading Analog Watch Faces with Simple CNN Heads

Omri Nidam

omrinidam@mail.tau.ac.il

Tel Aviv University

Introduction

Reading time from analog watch images requires precise geometric reasoning. A model must localize the clock hands, estimate their angles relative to the dial, and convert those angles into discrete hour and minute readings (every six degrees corresponds to one minute step). While recent vision–language systems excel at open-ended descriptions, they tend to underperform on this kind of fine-grained, geometry-sensitive recognition. Minute prediction is especially fragile when the minute hand lies near a tick mark or overlaps with the hour hand. This task exemplifies a class of visual problems where tiny spatial differences lead to different semantic outcomes—common in reading analog gauges or instruments where precision matters.

In this project, we study the clock-reading task using the Synthetic Watch Faces dataset. We first measure how a strong open vision–language model performs when prompted directly, establishing a baseline. We then propose a deliberately simple, task-specific approach: a ResNet-18 backbone as a feature extractor with two lightweight classification heads (one for hour and one for minute). Training proceeds in two phases for stability and efficiency: an initial stage with the backbone frozen and only the heads trained, followed by a short fine-tuning stage that unfreezes the last residual block to adapt high-level features to dial structure and hand geometry. Small implementation choices prove important for reliability and accuracy, so we use a modest input resolution, restrained data augmentation that does not corrupt labels, and a loss that slightly emphasizes the minute prediction. The result is a compact model that runs quickly, yields substantially higher accuracy than the prompted baseline (on exact time as well as hour-only and minute-only metrics), and produces interpretable results that clearly show its success and failure modes.

Related Work

This project lies at the intersection of prompt-based multimodal models and supervised, task-specific visual recognition. Large vision–language models have demonstrated impressive breadth in open-ended visual tasks, but they often struggle with tasks requiring precise localization or measurements. Reading analog gauges, dials, or clocks is a prime example: a few pixels of hand movement can change the reading, which frequently confuses generalist models. Classical computer vision approached such problems with geometric techniques—detecting the circular dial, finding line segments for hands, and computing angles explicitly. Modern deep networks can learn these subtleties given supervision, but without explicit training for fine-grained outputs, even advanced VLMs can output incorrect times.

In contrast, compact convolutional backbones pretrained on large image datasets provide stable mid-level features that can be adapted to structured targets like hours and minutes with small heads. Prior work on domain-specific reading tasks (e.g. analog meters and instrument panels) found that lightweight supervised models often outperform generic VLMs when the required label granularity is high and visual differences are subtle. Our approach follows this line of thought: we keep the backbone small and reliable, shape the prediction heads to the structure of the labels, and fine-tune minimally to capture the dial's geometry without overfitting.

Data

We use the Synthetic Watch Faces dataset (from Hugging Face), which consists of 11,000 rendered images of analog watch faces with corresponding time labels. The dataset is divided into standard splits:

- **Train:** 8,000 images covering 90% of all possible hour:minute combinations (the main set of times).
- **Validation:** 1,000 images drawn from the same distribution as Train.
- **Test:** 1,000 images drawn from the same distribution as Train.
- **Test Novel:** 1,000 images covering the held-out 10% of combinations (times never seen during training).

Each image is paired with a label in the format "HH:MM" indicating the time shown on the watch. For training, we convert the hour and minute from the string label into numeric class indices (hours 1–12 mapped to classes 0–11, and minutes 00–59 mapped to classes 0–59).

For evaluation, we compute three accuracy metrics:

- **Exact time accuracy:** the predicted hour and minute both match the label (correct HH:MM).
- **Hour-only accuracy:** the predicted hour matches the label hour (ignoring the minute).
- **Minute-only accuracy:** the predicted minute matches the label minute (ignoring the hour).

This breakdown highlights where the task is most challenging: minute-only accuracy is typically much lower than hour-only accuracy, reflecting how sensitive minute classification is to small angular errors.

We ensure a clean evaluation protocol in all experiments. Models train on the Train split and are validated on the Val split, with no peeking at Test or Test Novel until final evaluation. In this report we primarily report validation results (and note test results if available) to choose models consistently. To assist qualitative analysis, we also set aside a small "reference gallery" of images from the test splits for visualizing predictions, ensuring these images are not used during training.

For preprocessing, images are resized to a moderate resolution (we use 224×224 pixels for training) and normalized with the standard ImageNet mean and standard deviation (since our backbone was pretrained on ImageNet). We apply conservative data augmentation: a slight color jitter to simulate lighting changes, but no geometric transformations that would alter the time (we avoid random rotations or flips because they would require adjusting the labels accordingly). This restrained augmentation proved important for stable minute prediction. During a fine-tuning phase, we increase the input size to 256×256 to give the model a higher-resolution view of the clock face; this helps it resolve fine details like tick marks at the dial's perimeter, which can improve minute reading accuracy.

Methods

Our model is deliberately simple. We use ResNet-18 (pretrained on ImageNet) as a fixed visual feature extractor, and attach two fully connected layers as classification heads: one for the hour (12-way classification) and one for the minute (60-way classification). The model outputs an "HH:MM" prediction by taking the argmax class from each head and formatting them as hour and minute.

Training strategy: We train the model in two phases. In the first phase, we freeze all weights of the ResNet-18 backbone and train only the two heads. This provides a stable start, leveraging the general features learned during ImageNet pretraining while quickly fitting the classifier layers to the clock reading task. In the second phase, we unfreeze a portion of the backbone for fine-tuning. Specifically, we unfreeze the last residual block (and in some trials the second-last block as well) and continue training both the backbone and heads together for a few epochs. By only fine-tuning the top layers of the network (rather than the entire backbone), we allow the model to adjust to watch-specific patterns (like the layout of dials and the appearance of hands) without overfitting or requiring lengthy training. This focused fine-tuning injects just enough capacity to significantly improve accuracy on the task.

Loss and optimization: We use a joint loss that sums the cross-entropy for the hour prediction and the cross-entropy for the minute prediction. Because minute prediction is more fine-grained, we give it a slightly higher weight than the hour loss (e.g. a 2:1 ratio) to ensure the model prioritizes minute accuracy during training. We also apply a small amount of label smoothing to each softmax output to prevent overconfidence. Optimization is done with AdamW. We use a higher learning rate for the newly initialized head layers and a lower learning rate for the pretrained backbone layers (when they are unfrozen) to avoid destabilizing those features. Training is performed for only a few epochs in each phase, with a modest batch size, so the entire process is efficient on modest hardware.

Inference: At inference time, the model processes an input image through the ResNet backbone and the two heads, then simply selects the highest-probability hour class and minute class. We do not use any special post-processing beyond this argmax prediction.

We implemented the model in PyTorch. To ensure reproducibility of results, we fixed random seeds for initialization and data loading and saved the trained model checkpoints and logs for each experiment.

Experiments

We conducted experiments to compare the prompted vision–language model baseline against our specialized CNN model, and to evaluate the impact of each training stage. Table 1 summarizes the accuracy results of the different approaches on the validation set.

Table 1: accuracy results

Model/Approach	Exact	Hour-only	Minute-only
Prompted VLM (Qwen2-VL-2B baseline)	0.2%	12.6%	0.6%
CNN (ResNet-18 + heads, frozen backbone)	0.8%	19.8%	4.8%
CNN (ResNet-18 + heads, fine-tuned backbone)	14.9%	77.6%	18.5%
Ablation: + alignment pre-processing step	6.2%	30.7%	11.9%

Baseline – Prompted VLM: We first evaluated a large vision–language model to see how a generalist system handles this task. We used the Qwen2-VL-2B model (a ~2 billion parameter multimodal transformer) with a prompt instructing it to output the time in "HH:MM" format given the watch image. The model performed very poorly: on a sample of 500 validation images, it achieved essentially **0%** exact time accuracy (Table 1). It seldom got the hour correct (only about 12.6% hour-only accuracy) and almost never got the exact minute (0.6% minute-only). This confirms that a generic VLM, even a very large one, struggles with the precise geometric reading required for telling time on an analog clock. The VLM might not have been explicitly trained for reading clocks, and its language-based decoding tends to produce approximate or incorrect answers when exactness is needed.

Supervised CNN (frozen backbone): Next, we trained our task-specific CNN. After the first phase (frozen backbone, trained heads), the model already surpassed the VLM's performance. It achieved around **0.8%** exact accuracy on the validation set, with **19.8%** hour-only and **4.8%**

minute-only accuracy. These numbers are still low in an absolute sense, but they indicate the model learned some basic ability to read the clock (especially the hour hand) using the pretrained features. The hour-only accuracy of ~20% is far above random chance (8.3% for 12 classes), showing the model can often determine the correct hour even without any backbone fine-tuning. Minute accuracy remained very low at this stage, underscoring the need for further fine-tuning to handle the subtle visual differences between minute positions.

Supervised CNN (fine-tuned backbone): After fine-tuning the backbone's last block and increasing the input resolution to 256×256, the CNN's performance improved dramatically. The model reached **14.9%** exact time accuracy on validation. Its hour-only accuracy jumped to **77.6%**, indicating it was getting the hour correct on most images. Minute-only accuracy also climbed to **18.5%**, a significant improvement over the earlier phase. This demonstrates that adapting the high-level features to the specifics of clock geometry and providing more input detail had a huge impact on the model's ability to parse minutes. Compared to the baseline, our final model's exact accuracy is orders of magnitude higher, and even the minute prediction alone is about 30 times better (0.6% vs 18.5%). The majority of errors in the fine-tuned model now involve minutes: often the model's predicted minute is off by a small amount (for example, predicting 10:58 for an image showing 10:59, or similar near-misses). These errors usually occur when the minute hand is very close to a tick mark or when the hands overlap, making the exact reading ambiguous. Nonetheless, the overall performance indicates a clear practical advantage of the specialized CNN approach for this task.

Ablation – Alignment pre-processing: We also tried a simple alignment heuristic before feeding images to the model, with the aim of simplifying the problem. The idea was to rotate each image such that the minute hand would point straight up (12 o'clock position), effectively normalizing the orientation. We attempted this by detecting the longest line in the image (as a proxy for the minute hand) and rotating the image accordingly. In practice, this approach was not reliable: the heuristic sometimes picked the wrong line (e.g., the hour hand or a background element), resulting in incorrect rotations. When we evaluated the model on validation data with this pre-processing, performance actually dropped (Table 1 shows only **6.2%** exact accuracy with alignment, versus 14.9% without). The misalignment in those cases confused the model more than it helped. As a

result, we abandoned this alignment step in the final pipeline. This ablation reinforces that our end-to-end learned approach was more dependable than adding a brittle pre-processing step.

In summary, our experiments show that a simple CNN trained on this task vastly outperforms a large general-purpose VLM in reading analog clocks. The two-phase training (with careful augmentation and fine-tuning) was crucial to boost the minute hand recognition. We also examined the model's outputs qualitatively: the VLM baseline often produced arbitrary or default times (e.g. many outputs of "12:00" regardless of input), whereas our CNN's mistakes were more reasonable (often just a few minutes off, or an hour off when the hands were in between hours). This indicates the CNN learned a meaningful representation of clock geometry.

Conclusion

A compact, task-specific model can read analog watch faces far more accurately than a prompted generalist model. Our solution — a ResNet-18 with two classification heads trained in a targeted way — delivered a strong jump in exact time recognition and vastly improved hour and minute accuracies. The main remaining challenge is fine minute discrimination in edge cases (e.g. a minute hand near a boundary or overlapping with the hour hand), but even in these cases the model's errors are typically small.

There are several directions for further improving this clock-reading system:

- **Label-aware augmentation:** Incorporate rotations or flips during training while adjusting labels accordingly. For example, rotate an image by a random few degrees and shift the minute label by the same amount, or flip the image and adjust the time reading if necessary. This can increase data variety without ever corrupting the labels.
- **Higher resolution & deeper fine-tuning:** Train with higher input resolutions (beyond 256×256) for even more detail, and consider unfreezing an additional layer block of the backbone (with a very low learning rate) to allow the model to learn more specialized features. These steps typically yield a few extra points of accuracy by providing more capacity and clarity on small details.

- **Structured minute prediction:** Change the minute prediction to exploit its circular nature. For example, the model could predict a coarse 5-minute bin alongside a fine offset, or output a pair of values (sine and cosine of the minute angle) and train with a circular loss. Such approaches could reduce errors around the 59↔00 minute boundary by making the model aware that those two are adjacent in time.
- **Geometric priors:** Integrate simple geometric reasoning into the model. One idea is to have an auxiliary output that detects the clock's center or the orientation of the dial, or to provide the model with an extra input channel highlighting the clock border or center. Another idea is to use a classical vision technique to find candidate line segments for hands and feed that information (perhaps as a heatmap) into the network. Injecting these priors could guide the model, especially in cases where the hands overlap or have similar shapes.
- **Expanded synthetic training data:** Since the dataset is synthetic, we can easily generate more examples. Rendering additional watch faces with varied designs, backgrounds, lighting, slight blur, or different random rotations (with labels adjusted accordingly) could improve robustness. The model could also be fine-tuned on a small set of real-world clock images (if available) to ensure it generalizes beyond the synthetic domain.

In conclusion, our work demonstrates that a small, well-tuned CNN can significantly outperform a large general-purpose VLM on a fine-grained visual task like reading analog clocks. By leveraging pretrained features and focusing on task-specific details through a two-phase training process, we achieved a level of precision that the general model could not match. This fast and straightforward model sets a strong baseline for analog gauge reading. With the proposed extensions, it could be improved further and applied to other related tasks where exact visual interpretation is required.

Code

https://github.com/omrinidam18/analog-watch-reading.git