# It's About Time: Analog Clock Reading in the Wild

Charig Yang[1]        Weidi Xie[1,2]        Andrew Zisserman[1]

[1]VGG, Department of Engineering Science, University of Oxford [2]Shanghai Jiao Tong University

{charig,weidi,az}@robots.ox.ac.uk
https://www.robots.ox.ac.uk/~vgg/research/time

## Abstract

*In this paper, we present a framework for reading analog clocks in natural images or videos. Specifically, we make the following contributions: First, we create a scalable pipeline for generating synthetic clocks, significantly reducing the requirements for the labour-intensive annotations; Second, we introduce a clock recognition architecture based on spatial transformer networks (STN), which is trained end-to-end for clock alignment and recognition. We show that the model trained on the proposed synthetic dataset generalises towards real clocks with good accuracy, advocating a Sim2Real training regime; Third, to further reduce the gap between simulation and real data, we leverage the special property of "time", i.e. uniformity, to generate reliable pseudo-labels on real unlabelled clock videos, and show that training on these videos offers further improvements while still requiring zero manual annotations. Lastly, we introduce **three** benchmark datasets based on COCO, Open Images, and The Clock movie, with full annotations for time, accurate to the minute.*

## 1. Introduction

Humans are able to sense the time to some level of granularity given environmental cues, such as luminance or the extent of shadows. However, in order to know the *exact* time we read from a time-keeping instrument such as a clock or watch. Clocks come in different shapes, forms and styles, and humans are able to read them despite not having seen the particular clock before. In this paper our objective is to enable a machine to perform the same task of telling the time from clocks *in the wild*.

Nowadays, clocks come in two main types – digital and analog. Digital clocks can be handled by text spotting methods [25, 27–29, 32] with relative ease, which we show in the Appendix, but reading analog clocks is a different and challenging problem: there are significant appearance variations between clock faces (see Figures 2 and 5), their im-

aged shape and the position of the numbering is severely affected by camera viewpoint, and the presence of shadows and specular reflections add confusion with the clock hands. While this problem has existed for a long time [1, 9, 35], no previous solutions are able to robustly read the time from clocks, apart from under extremely limited situations. And, somewhat surprisingly, reading analog clocks in unconstrained images has been largely overlooked in the computer vision literature. Additionally, there are no reliable benchmarks for evaluation, hindering the research community from tackling this task.

However, there are similarities between analog clock reading and text spotting in natural scenes – since in both cases the design (of the clock face or text font) is chosen to be readable, and in both cases there is a detection stage and then a reading stage. Clock reading has the additional challenges outlined above, but it also has an additional redundancy cue in that the position of the hour hand gives some information about the position of the minute hand. Given the similarities between the two tasks, we start with an approach that has been successful for text spotting: using synthetic datasets [12, 18] and spatial transformer networks [34], and ask if these ideas transfer to our task. We find that they do to an extent, and provide further contributions to bridge the Sim2Real generalisation gap.

While being able to carry out a new task is a sufficient reward in itself, there are a number of applications that are opened up, once we are able to automate time reading in images in the wild: first, it will now be possible to offer corrections where the image's EXIF metadata differs from the time read in the image; second, in video forensics, it will now be possible to spot if the video has been tampered with if the temporal ordering does not progress monotonically or if there is manipulation of the speed [15]; third, it provides a new method of searching, retrieving and grouping images and videos; and, finally, clocks are just a (rather difficult) instance of an analog scale, and the methods we have proposed can be applied with a simple adaptation to other type of scales – from scientific instruments to industrial gauges.

In this paper, we provide the first working solution to

these issues. We make the following contributions:

*First*, we propose a synthetic dataset generator, Syn-Clock, that is designed to generalize to real clocks. Syn-Clock has several controllable features that enables it to generate clocks with a wide range of designs. Moreover, we mimic difficulties faced in recognising real clocks into the generator's data augmentation process, *e.g.* homography transformation, artefacts, shadows.

*Second*, we design a two-stage framework involving detection and recognition stages. The detection can simply be an off-the-shelf object detection model. The recognition stage involves an alignment network, which is a spatial transformer network that regresses homography transformation parameters in order to make the clock fronto-parallel, and a classification network, which determines the time accurate to the minute. We show that the model is able to generalise towards real clocks with good accuracy.

*Third*, we leverage the uniformity of time – that it flows at a constant rate, in order to generate pseudo-labels on unlabelled clock videos. Specifically, we can be reasonably confident that the time labels in a video are correct if the rate of change of predicted time is constant throughout the video. To achieve this, we use a bundle adjustment algorithm to filter eligible videos and train on those with the pseudo labels. We also propose a dataset of 3,443 unlabelled clock time-lapse videos, and show that learning from pseudo-labelled real data improves the performance. We will release the raw videos, as well as reliable automatic annotations for 1.5M frames across 2,511 videos.

*Fourth*, we propose three new benchmark datasets. The first two are based on existing datasets for object detection, namely COCO and OpenImages. We also introduce the Clock Movies dataset, based on the film The Clock (2010), which is a 24-hour montage of different movies featuring clocks[1]. Our model achieves 80.4%, 77.3% and 79.0% top-1 accuracy on each dataset respectively, marking the first time that analog clocks can be read successfully in unconstrained images.

## 2. Related Work

**Analog clock reading.** While there exists blog posts and repositories, this task is still largely absent in the research literature, which may be because of the lack of proper benchmarks. Traditional methods use handcrafted methods such as edge or line detection algorithms to read clocks [10, 14, 16], but they only work on simple, clean, artefact-free, fronto-parallel clocks. Given the absence of labelled data, existing works usually consider to train on synthetic clocks [1, 9, 35], but only go as far as testing their models on synthetic test sets. These models hence are not

---

[1]Due to copyright restrictions, we may not be able to release the frames for the Clock Movies dataset.

designed with generalisation in mind, and usually fail to read clocks in real scenes. In this paper, in addition to adopting a more challenging data simulation pipeline, we also leverage the uniformity property of time, and mix the training with data of timelapse analog clock videos. As a result, the gap between simulation and real-world images has been largely minimised in a self-supervised manner, *i.e.* without using manual annotations.

**Dial reading.** The closet kin to our work is on automatic dial or gauge meter readings. These two tasks face similar challenges in overcoming the effects of blur, glare, and reflections. The solutions proposed are somewhat similar, using neural networks [2], projective transforms [4], and virtual dataset generators [6, 17], which work very well for gauges with known shape and style. These have applications towards reading electricity, water or gas dials [2, 23, 31, 33]. Our task is more challenging as clock appearances vary greatly, and we further propose a method to bridge this generalisation gap.

**Sim2Real transfer.** In many computer vision and robotic tasks, synthetic datasets prove to be a useful source for training. This is especially true when the ground-truth is difficult or impossible to acquire at scale, including optical flow [5, 30], and text detection [12] and recognition [18], pose estimation [8], and motion segmentation [22].

## 3. Architecture

In this paper, our goal is to train a computer vision system that can read the time shown on the analog clock from in-the-wild images. To achieve this, we propose a framework shown in Figure 1. Specifically, our proposed architecture takes an image as input, then sequentially undergoes *cropping*, *alignment*, and *reading*.

### 3.1. Clock Localisation Module ($\Phi_{\text{loc}}$)

Given an input image $\mathcal{I}$, we pass it through an object detection network, to localise and crop the clock.

$$\mathcal{I}_{\text{crop}} = \Phi_{\text{loc}}(\mathcal{I}; \Theta_{\text{loc}}) \in \mathbb{R}^{3 \times h_c \times w_c} \tag{1}$$

As we will show in the experiments in Section 7, the detection stage can be done using an off-the-shelf object detector [24] without much impact on performance. This work hence focuses on the recognition stage.

### 3.2. Clock Recognition Module ($\Phi_{\text{rec}}$)

The cropped clock could be passed directly to a classification network. However, this is usually not ideal for two reasons, firstly, due to the imperfections of the localisation module; and secondly, even if the clock is properly localised and cropped, it can still sometimes be difficult to read due to the viewpoint (as viewpoint alters
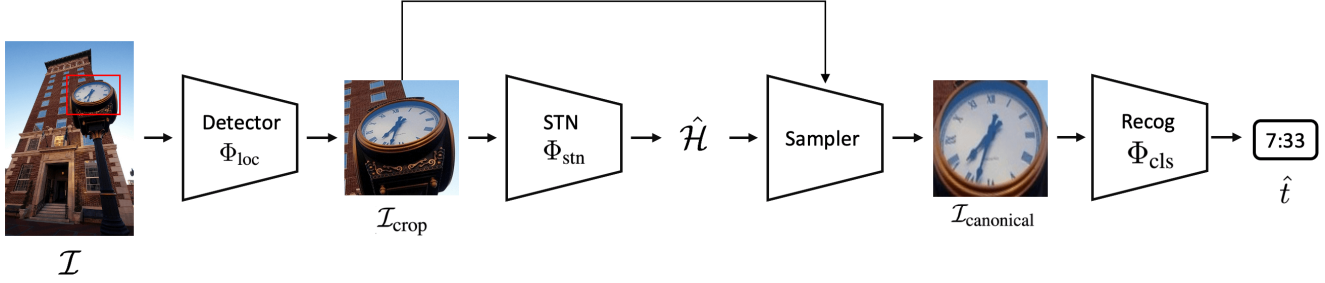
Figure 1. **Architecture.** Given an image $\mathcal{I}$, we first use an off-the-shelf object detector $\Phi_{\text{loc}}$ to obtain a cropped image $\mathcal{I}_{\text{crop}}$. We then pass the cropped image to a spatial transformer network $\Phi_{\text{stn}}$ which outputs a homography matrix $\mathcal{H}$, that can then be used to warp the cropped image to a canonical view $\mathcal{I}_{\text{canonical}}$. Lastly, the canonical image is passed to a classification network $\Phi_{\text{cls}}$ to predict the time.

angles). To overcome these problems, we adopt a spatial transformer network (STN) [19] for *alignment*, facilitating the *recognittion*, *i.e.* $\Phi_{rec}(\cdot) = \Phi_{\text{cls}}(\Phi_{\text{stn}}(\cdot))$.

**Spatial Transformer Network ($\Phi_{\text{stn}}$)** takes the cropped image ($\mathcal{I}_{\text{crop}}$) as input and outputs 8 homography transformation parameters:

$$\hat{\mathcal{H}} = \Phi_{\text{stn}}(\mathcal{I}_{\text{crop}}) \in \mathcal{R}^{3\times 3} \qquad (2)$$

$$\mathcal{I}_{\text{canonical}} = \text{SAMPLER}(\mathcal{I}_{\text{crop}}, \hat{\mathcal{H}}) \in \mathcal{R}^{3\times h\times w} \qquad (3)$$

where $\hat{\mathcal{H}} \in \mathcal{R}^{3\times 3}$ refers to the predicted homography transformation with 8 degree of freedoms, and $\text{SAMPLER}(\cdot)$ denotes a differentiable warping, that transform the cropped clock to its canonical view ($\mathcal{I}_{\text{canonical}}$), where the clock is fronto-parallel and the '12' position is at the top.

**Classification Network ($\Phi_{\text{cls}}$).** For reading the time, we quantise the time and pose the recognition problem as a 720-way classification, *i.e.* classifying both the hour (12) and minute (60) together. Specifically, we pass the canonical clock ($\mathcal{I}_{\text{canonical}}$) to a classification network, which outputs the probability for each class.

$$\hat{t} = \Phi_{\text{cls}}(I_{\text{canonical}}) \in \mathcal{R}^{720} \qquad (4)$$

In the Appendix we compare this classification approach to time reading to a regression approach.
Each of the networks ($\Phi_{\text{stn}}$, $\Phi_{\text{cls}}$) is a standard ResNet-50 [13] pretrained on ImageNet [7].

After introducing the architecture, one question remains: *how can we efficiently train this clock recognition module, without a laborious annotation procedure?*

## 4. Synthetic Data and Sim2Real Training

To avoid laborious manual annotations, we describe a procedure for training alignment and recognition with simulated clocks, advocating a Sim2Real training regime. Specifically, the synthetic clocks are generated with different viewpoints, time, and styles.

### 4.1. Synthetic Clock Generator (SynClock)

Inspired by the idea for text spotting [18], we propose a scalable pipeline for generating synthetic analog clocks. In order to generalise towards real clocks, we make the dataset sufficiently diverse with several controllable parameters, and add artefacts to simulate the real-world scenario. Specifically, we vary these parameters during training:

- **Background**: color.
- **Clock face**: size, shape, color.
- **Clock border**: thickness, color.
- **Tick marks**: whether to have ticks every minute or every hour, gap from border, length, thickness, color.
- **Numerals**: whether to have numerals, gap from border, font, font size, font thickness, color.
- **Hands**: whether to use 2, 3, or 4 hands (hour, minute, second, alarm), time, length, back length, thickness, color, whether to use arrow, arrow tip length, arrow size. (may be different for each hand)
- **Artefacts**: shadows beside a hand, random lines, random homography transformation.
- **Augmentation**: random blur, color channel jittering

Figure 2 shows examples of clocks produced by this SynClock generator, and further examples are given in the Appendix.

### 4.2. Training on SynClock

With full controls on the data generation procedure, we are able to to train both spatial transformer and the classification network. Specifically, $\Phi_{\text{stn}}$ and $\Phi_{\text{cls}}$ are trained with L1 loss and cross-entropy loss, using ground-truth transformation ($\mathcal{H}$) and time ($t$) generated from SynClock:

$$\mathcal{L} = \mathcal{L}_{\text{stn}} + \mathcal{L}_{\text{cls}} = \sum |\hat{\mathcal{H}} - \mathcal{H}| + \sum \hat{t}\log(t) \qquad (5)$$

## 5. Pseudo-labelling Real Videos

While training on carefully designed synthetic clocks allows reasonable generalisation to real images, the domain

Figure 2. **Training data.** Left: example images from the SynClock dataset generator, which is designed to generate a wide range of clocks. We also add artefacts, such as random lines and shadows followed by augmentations. Right: example scenes from the Timelapse dataset, containing 3,443 unlabelled timelapse videos containing clocks. We train with this dataset using pseudo-labels with uniformity constraints.

gap between simulation and real images still exists, as will be demonstrated in our experiments in Section 7.3. In this section, we describe a simple idea for minimising the domain gaps by mixing the training with real video frames. One critical issue would be, *how can we obtain the ground-truth time for these video frames?*

## 5.1. Uniformity Constraints

We leverage the **uniformity** property of time, *i.e.* time flows unidirectionally at a constant rate in videos. Specifically, given an unlabelled clock video, we pass each individual frame through the localisation and recognition module to get the time predictions. As the recognition module has only been trained on synthetic data, it is likely to generate incorrect predictions for certain frames. Here, we can conveniently fit a line with RANSAC [11] – a robust fitting algorithm that allows line fitting in the presence of outliers, incorrect time predictions in our case. An iteration of RANSAC involves fitting a line with the minimal number of randomly sampled points (two), counting points within a threshold distance as 'inliers', and the points that are distant from the line are treated as outliers. After running RANSAC multiple times, the best fitted line with maximum number of inliers are adopted, and all outliers can be re-calibrated accordingly, as shown in Figures 3 and 4.

Note that the data is cyclic, *i.e.* 11:59 (719) is connected to 0:00 (0). To overcome this, we modified RANSAC such

that it fits a sawtooth wave instead of a line, as seen in Figure 3. Specifically, after fitting a line through the randomly sampled points, if the predicted time is outside the [0, 720) range, we have to repeatedly add or subtract it by 720 so that it falls into the valid range, before counting the inliers. In practice, this can be implemented simply by applying a *modulo* (%) 720 operator on the fitted line.

## 5.2. Timelapse Dataset

While the above method can work on any video with clocks running continuously, storing the videos of multiple hours long at scale would be impractical. Instead, we use *timelapse* clock videos, where time moves much faster, allowing video duration to be in the range of seconds and not hours. We hence collect a dataset of 3,443 unlabelled time-lapse clock videos from the internet. Although no absolute time information can be extracted from these unlabelled videos, knowing that the speed is constant is sufficient for our use.

**Joint Training.** After pseudo-labelling the videos, we select the ones with inlier proportion being above a threshold (fixed at 0.7), with a tight inlier margin of +-3 minutes. We also reject videos that move too slowly, that is, less than 10 minutes throughout the video. We add the videos that pass filtering to the training set to re-train the model. This whole process is automated, and hence no extra manual annotation or screening is required. Although the groundtruth
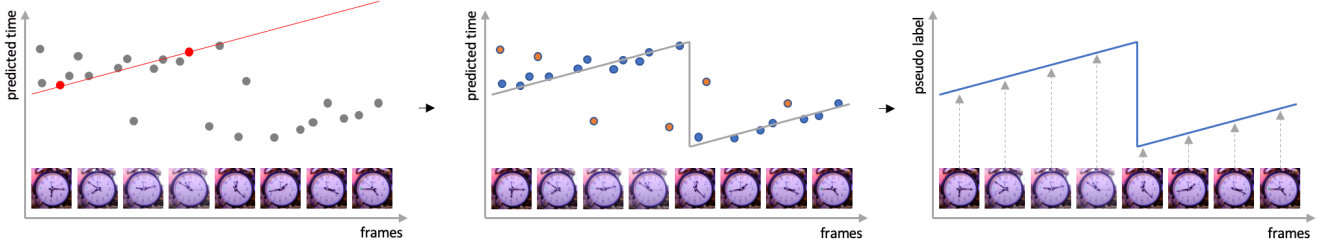
Figure 3. **Uniformity Constraints.** Given a timelapse video of a clock, we iteratively fit a line through randomly sampled predictions (left), before rectifying it into the valid range [0, 720) using the modulo operator and counting the inliers (middle). If the maximum inlier count is above a threshold, we then correct the readings using the fitted line (right) and add the pseudo-labelled clocks to the training set.
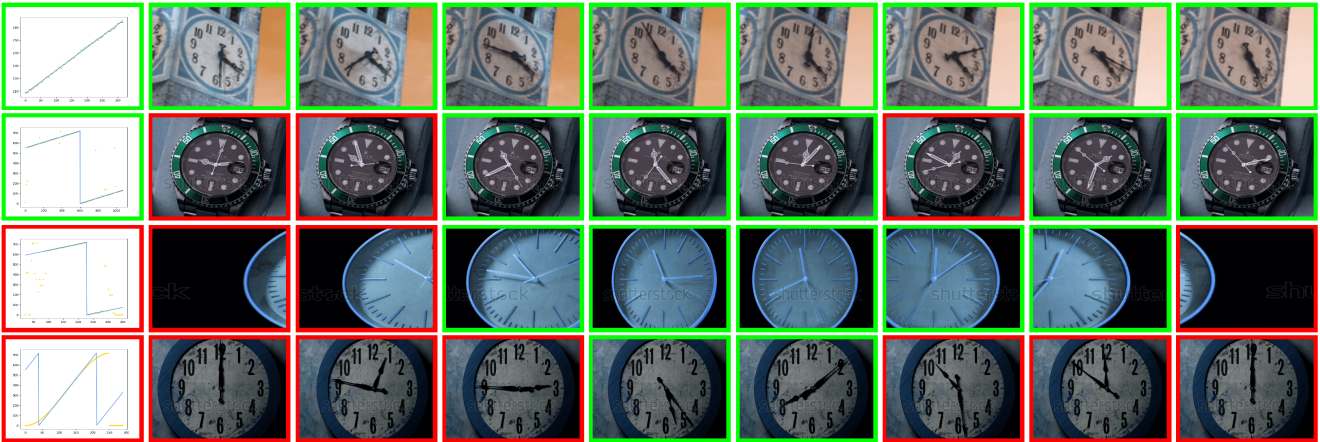


Figure 4. **Examples of filtering using uniformity constraints.** The first two rows show examples of videos that passed filtering, with incorrect predictions being calibrated accordingly. The bottom two rows show examples of videos that failed, due to out-of-frame and non-uniform speed respectively. The color of the box indicates success or failure.

transformation for aligning the clocks remains unavailable, meaning the intermediate regression loss only applies to the synthetic data, the model is still differentiable end-to-end.

$$\mathcal{L} = \mathcal{L}_{\text{stn}} + \mathcal{L}_{\text{cls}} = \sum_{\text{SynClock}} |\hat{\mathcal{H}} - \mathcal{H}| + \sum \hat{t}\log(t) \quad (6)$$

**Iterative retraining.** The process of pseudo-labelling and retraining can be performed iteratively. Specifically, after training with real data, the model becomes more capable of reading clocks, and hence can be used to generate better pseudo-labels to train the model. This process can be iterated for further performance improvements.

# 6. Experimental Setup

## 6.1. Datasets

We use separate datasets for training and testing. We utilise one synthetic and one pseudo-labelled real dataset for training, and hence are able to train the model with *zero* manual annotations. As this is a new task, there has not been proper benchmarking in the literature. We therefore propose three different test datasets. The summary statistics

are given in Table 1. **All** five of these datasets are contributions of this paper.

| Dataset | Type | Size | Split | Bbox | $\mathcal{H}$ | Time |
|---------|------|------|-------|------|---------------|------|
| SynClock | image | $\infty$ | train | ✗ | ✓ | ✓ |
| Timelapse | video | 3443 | train | ✗ | ✗ | ✗ |
| COCO | image | 1911 | test | ✓ | ✗ | ✓ |
| OpenImages | image | 1317 | test | ✓ | ✗ | ✓ |
| Clock Movies | image | 1244 | test | ✗ | ✗ | ✓ |

Table 1. **Statistics of the proposed datasets.** The first two datasets are used for training, while the other three for testing.

### 6.1.1 Training

**SynClock.** We train our recognition module using the SynClock dataset, as previously explained. We also provide ground truth homographies to train the alignment network.
**Timelapse.** This dataset is a subset of WebVid dataset [3] used for video retrieval, using the "clocks time lapse" as the keyword for search query. It contains 3,443 unlabelled videos, and we train on this dataset using pseudo-labels, with the number of filtered videos shown in Table 2.

(a) COCO          (b) OpenImages          (c) Clock Movies

Figure 5. **Evaluation data.** From left to right, the figure shows example scenes and cropped clocks from COCO [26], OpenImages [21] and Clock Movies datasets. For the first two datasets, we filter scenes with readable analog clocks from the original datasets and provide time labels, while the latter is from 600 different movies. They contain 4,472 images in total and are used for evaluation.

| Iteration | Videos | Frames |
|---|---|---|
| Pseudo-label Round 1 | 1670 | 1.02M |
| Pseudo-label Round 2 | 2052 | 1.30M |
| Pseudo-label Round 3 | 2398 | 1.46M |
| Pseudo-label Round 4 | 2460 | 1.48M |
| Pseudo-label Round 5 | 2511 | 1.51M |

Table 2. **Statistics of the Timelapse dataset.** This table shows the number of videos successfully labelled in each iteration of pseudo-labelling. Each iteration uses the model of the previous row.

### 6.1.2 Testing

**COCO.** This dataset is a subset of COCO [26] dataset that contains clocks, with bounding boxes provided for 1,911 images and time manually labelled by the authors.

**OpenImages.** This dataset is a subset of OpenImages [21] dataset that contains clocks, with bounding boxes provided for 1,311 images and time manually labelled by the authors.

**Clock Movies.** This dataset is based on the film The Clock (2010), which is a 24-hour film montage of different movies featuring clocks. We collect 1,244 images from over 600 different movies from the movie's Fandom page. As the timestamp within the movie reflects the absolute time by design, the time label can be implicitly obtained.

## 6.2. Implementation details

For the localisation stage, we use an off-the-shelf detector CBNetV2 [24] to crop the clocks, and add 20% context to each side of the image to ensure recall. The detector is trained on COCO [26] and achieves the state-of-the-art performance on COCO among methods with publicly released models at the time of writing.

We first train the recognition model on SynClock dataset. During training, the dataset is generated on-the-fly. We train the model using the Adam [20] optimiser with learning rate 1e-4 and batch size 32 for 100k iterations.

We then use the model trained on SynClock to generate pseudo-labels for the Clock Time-lapse dataset, and filter valid ones into the training set using uniformity constraints. We then fine-tune the trained model on the enlarged training set for 20k iterations with batch size 64, with half being from SynClock and half being from the pseudo-labelled dataset. We then repeat the process where we use the newly trained model to generate pseudo-labels, and then retrain the model using the same settings as the first retraining.

## 6.3. Evaluation metrics

As this is a new task, there is no existing method for evaluation. We therefore propose the following metrics:
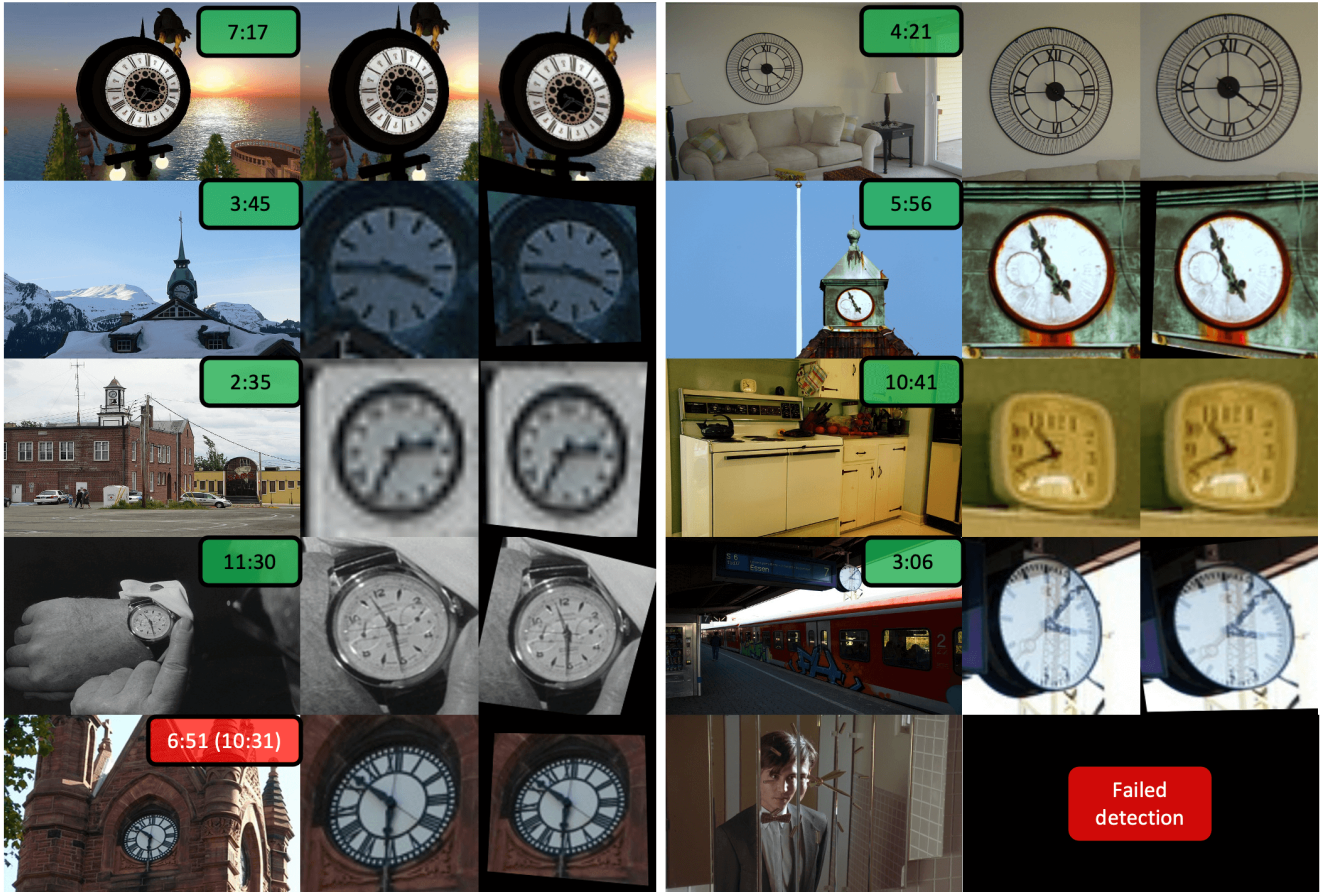
Figure 6. **Qualitative results.** The columns show the original, cropped, and canonical images, with predicted time overlaid on the original image. The model is able to read clocks in varying and challenging scenes, including those with different styles (rows 1-2), low resolution (row 3), and non-frontal viewing angles (row 4). The bottom row shows two failure cases: reading swapped hands and failed detection.

- **Hour accuracy.** when the predicted hour is correct.
- **Minute accuracy.** when the predicted minute is correct within a +-1 margin. We think this is close to the limit of human readability in resolving the in-between cases, especially with the absence of the second hand.
- **Overall accuracy.** when both hour and minute are correct (minute within a +-1 margin).

We also look into top-1, 2, 3 prediction accuracies, as some clocks have ambiguities with hands.

## 7. Results

### 7.1. End-to-end Results

For a successful detection, both localisation and recognition have to be successful. To our best knowledge, no existing work has been able to achieve success in this setting, so it is challenging to compare to prior work. The results for the full model are shown in Table 3. We show that the model is able to achieve success in all these challenging datasets. Note that the accuracy is lower than that

| Dataset | 1 | 1-H | 1-M | 2 | 3 |
|---|---|---|---|---|---|
| COCO | 80.4 | 86.9 | 84.4 | 84.3 | 86.5 |
| OpenImages | 77.3 | 83.5 | 81.9 | 81.5 | 84.6 |
| Clock Movies | 79.0 | 82.3 | 82.4 | 83.3 | 85.5 |

Table 3. **End-to-end results.** We report the accuracy on the three datasets. 1, 2, 3 are the top 1, 2, and 3 overall accuracies respectively, whereas 1-H and 1-M are the top-1 hour and minute accuracies. The model achieves success in all these datasets.

| Dataset | mAP | AP50 | AP75 |
|---|---|---|---|
| COCO-val | 77.0 | 92.1 | 74.6 |
| OpenImages | 61.5 | 89.2 | 66.4 |

Table 4. **Localisation results.** We report the average precision (AP) of the bounding box with respect to the ground truth. Note that the object detector itself is trained on COCO-train, so only its validation set is reported.

| Training | COCO (IOU>50) | | | | | OpenImages (IOU>50) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1-H | 1-M | 2 | 3 | 1 | 1-H | 1-M | 2 | 3 |
| SynClock | 12.2 | 25.9 | 27.1 | 16.5 | 19.4 | 13.8 | 28.8 | 26.2 | 17.5 | 20.9 |
| + Augmentation | 16.3 | 30.7 | 29.8 | 21.5 | 25.2 | 19.0 | 33.8 | 30.9 | 23.2 | 25.9 |
| + Homography | 47.8 | 58.4 | 58.6 | 57.1 | 61.9 | 44.2 | 54.6 | 56.5 | 51.5 | 56.5 |
| + Artefacts | 57.9 | 68.2 | 65.0 | 66.2 | 70.7 | 53.2 | 63.2 | 61.5 | 58.4 | 63.3 |
| + Spatial Transformer | 59.6 | 71.3 | 67.0 | 67.1 | 71.4 | 55.4 | 65.8 | 64.2 | 61.8 | 65.9 |
| + Timelapse Round 1 | 73.9 | 81.5 | 79.1 | 78.6 | 81.7 | 69.5 | 76.7 | 75.6 | 74.1 | 77.4 |
| + Round 2 | 80.2 | 86.3 | 84.6 | 83.9 | 86.5 | 75.8 | 81.7 | 81.3 | 80.1 | 83.1 |
| + Round 3 | 82.6 | 88.1 | 85.7 | 86.2 | 88.1 | 79.0 | 83.5 | 83.5 | 81.7 | 84.5 |
| + Round 4 | 82.8 | 88.4 | 86.1 | 86.4 | 88.6 | 78.9 | 83.6 | 83.5 | 82.4 | 85.5 |
| + Round 5 | 82.9 | 88.4 | 86.8 | 86.6 | 88.5 | 79.6 | 84.7 | 83.9 | 83.3 | 86.2 |

Table 5. **Recognition results**. Here, all evaluations are conducted on clocks of IOU>50 with respect to the groundtruth annotations, to isolate the effect of detection failure and focus solely on incorrect predictions from recognition failure. We show that augmentation, alignment (STN), and pseudo-labelling are all essential components for improving the performance of recognition module.

reported in the last row of Table 5, as this also includes the cases where the detection is unsuccessful.

## 7.2. Localisation-only Results

To disentangle the effects of localisation and recognition, we firstly report the results for localisation on COCO and OpenImages in Table 4. To be consistent with the object detection literature, we report the average precisions (AP50, AP75), where the bounding box IoU is over a threshold. We also report the mean average precision (mAP), which is the average of APs across the thresholds [50:5:95]. Overall, the detection task is reasonably successful.

## 7.3. Recognition-only Results

To evaluate the performance only on recognition, we only select images where the bounding box IoU is above 50%. Table 5 shows incrementally the effects of different components within the model.

**SynClock.** We show the effects of parts that constitute SynClock, namely data augmentation, homography transformation, and artefacts. The homography contributes the most towards accuracy (+31.5%/+25.2% for COCO/OpenImages top-1 accuracy respectively) as it allows clock reading from various angles and sizes. Augmentations (+4.1%/+5.2%) such as blurring and jittering and artefacts (+10.1%/+9.0%) such as shadows and random lines both help bridge the Syn2Real generalisation gap.

**Spatial Transformer.** We show the effect of the spatial transformer within the architecture, which results in a monotonic improvement across all metrics (+1.7%/+2.2%).

**Pseudo-labelling Videos.** We show that adding pseudo-labelled real video to the training set greatly contributes towards accuracy (+14.3%/+14.1%). Iteratively repeating the process also yields further improvements (+9.0%/+10.1%).

## 7.4. Qualitative Visualisation

We show the qualitative results in Figure 6, from localisation, alignment and recognition. Each process can potentially give errors, which will impact the performance. We show that our model generalises to various styles of clocks, and is able to overcome low-resolution and alignment problems. The failure case is also shown in the bottom row.

The case where the model reads swapped hands is a good example of the limitation of our model. In this example, the hands appear to be of similar length. We (humans) can resolve this ambiguity and tell the correct time (10:31) because we look at the relative position between the hour hand and the hour mark. That is, if the time were to be 6:51 (as the model incorrectly predicted), then the hour hand should be closer to 7, and not 6. The model is not yet able to reason to this level, and hence makes wrong predictions.

## 8. Conclusion

This work introduces a framework for clock reading in real images or videos. We also circumvent the lack of training data in the recognition stage by proposing the synthetic dataset generator SynClock, and iteratively pseudo-labelling on real unlabelled videos using uniformity constraints. Further, we propose three benchmark datasets with accurate time labels. In the future, clock reading should become a standard processing step for images, in the same way that text spotting and object detection are now.

# References

[1] Ahmed Al-Baz. Reading analog clocks with xception cnn network. https://github.com/albazahm/Reading_Analog_Clocks_with_Xception_CNN_Network, 2020. 1, 2

[2] Alexey Alexeev, Georgy Kukharev, Yuri Matveev, and Anton Matveev. A highly efficient neural network solution for automated detection of pointer meters with different analog scales operating in different conditions. *Mathematics*, 8(7):1104, 2020. 2

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 5

[4] Haojing Bao, Qingchang Tan, Siyuan Liu, and Jianwei Miao. Computer vision measurement of pointer meter readings based on inverse perspective mapping. *Applied Sciences*, 9(18), 2019. 2

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2

[6] Weidong Cai, Bo Ma, Liu Zhang, and Yongming Han. A pointer meter recognition method based on virtual sample generation technology. *Measurements*, 163:107962, Oct. 2020. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[8] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. In *NIPS*, 2019. 2

[9] Felix Duvallet. Deep time reading. https://github.com/felixduvallet/deep-time-reading, 2016. 1, 2

[10] Abdelrahman Osama El-Naggar, Hassan Mohamed Adam, Ziad Mahmoud El-Molla, and Hussein El-shobokshy. Convert from analog to digital clock. https://github.com/Abdelrahman-Elnaggar/Convert-from-analog-to-digital-clock, 2018. 2

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[12] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016. 1, 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[14] Balazs Horvath. Analog clock reading. https://github.com/aHorvathBazsi/Analog_clock_reading, 2021. 2

[15] Brian C Hosler and Matthew C Stamm. Detecting video speed manipulation. In *CVPRW*, 2020. 1

[16] Bassel Hossam. clockreader. https://github.com/basselhossam/clockreader, 2017. 2

[17] B. Howells, J. Charles, and R. Cipolla. Real-time analogue gauge transcription on mobile phone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021. 2

[18] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *NeurIPS Workshop*, 2014. 1, 2, 3

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 6

[22] Hala Lamdouar, Weidi Xie, and Andrew Zisserman. Segmenting invisible moving objects. In *BMVC*, 2021. 2

[23] Rayson Laroca, Alessandra B Araujo, Luiz A Zanlorensi, Eduardo C De Almeida, and David Menotti. Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach. *IEEE Access*, 9:67569–67584, 2021. 2

[24] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibing Ling. Cbnetv2: A composite backbone network architecture for object detection. *arXiv preprint arXiv:2107.00420*, 2021. 2, 6

[25] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE TPAMI*, 43(2):532–548, 2021. 1

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[27] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, 2018. 1

[28] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, 2020. 1

[29] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, 2018. 1

[30] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2

[31] Ayman Naim, Abdessadek Aaroud, Khalid Akodadi, and Chouaib El Hachimi. A fully AI-based system to automate water meter data collection in Morocco country. *Array*, 10:100056, 2021. 2

[32] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *ICCV*, 2019. 1

[33] Gabriel Salomon, Rayson Laroca, and David Menotti. Deep learning for image-based automatic dial meter reading: Dataset and baselines. In *IJCNN*, 2020. 2

[34] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, 2016. 1

[35] Victor Suarez Vara. Reading analog clocks with neural networks. https : / / github . com / VictorSuarezVara/Reading-analog-clocks-with-neural-networks, 2021. 1, 2