People@Places and ToDY: Two Datasets for Scene Classification in Media Production and Archiving

Werner Bailer[0000−0003−2442−4900] and Hannes Fassold[0000−0001−7113−0038]

JOANNEUM RESEARCH, Graz, Austria, {firstname.lastname}@joanneum.at

Abstract. In order to support common annotation tasks in visual media production and archiving, we propose two datasets which cover the annotation of the bustle of a scene (i.e., populated to unpopulated), the cinematographic type of a shot as well as the time of day and season of a shot. The dataset for bustle and shot type, called People@Places, adds annotations to the Places365 dataset, and the ToDY (time of day/year) dataset adds annotations to the SkyFinder dataset. For both datasets, we provide a toolchain to create automatic annotations, which have been manually verified and corrected for parts of the two datasets. We provide baseline results for these tasks using the EfficientNet-B3 model, pretrained on the Places365 dataset.

Keywords: datasets, neural networks, cinematography, video retrieval

1 Introduction

While research has moved from image classification to object detection, segmentation and other more advanced topics, performing classifications of images or entire shots of videos is still a practically relevant task in describing visual content in order to make it findable. This task occurs when describing newly arriving content for production purposes (e.g., news) or annotating large amounts of otherwise sparsely documented content in media archives. Locations are among the three most frequently used search facets in video archive search [13]. For many purposes in visual content creation, place categories (i.e., street, shopping mall) are needed rather than named locations. Automatically labeling images or video shots with such location categories is a typical classification problem, and the Places365 dataset [31] is a very well known resource for this task. However, in a practical setting, there are other key properties of the scene, that are relevant to judge whether a shot is usable or not.

First, it is important to know whether the scene is "empty", or there are people or vehicles visible. We call this property "bustle", i.e., whether there are traces of people being active in that scene or not. While it has always been an important query criteria to explicitly look for a quiet or busy view of the scene, the recent COVID-19 pandemic has made that a much requested feature, as depending on the level of restrictions valid at that time, news reports require either empty or populated street scenes.

Second, the shot type (sometimes called shot size) is a key cinematographic property, which determines the importance of a subject, and the context in which a particular shot can be used. The shot type is typically defined by the height ratio of the depicted persons in relation to the view.

Third, for outdoor shots the time of day and the season are important properties. A news editor searching outdoor shots of a building (e.g., house of parliament) wants to find shots that match the season of the story, as well as day or nighttime. For more scenic views, a sunrise or sunset shot is often requested.

Although these are not uncommon properties of content, there are hardly any datasets covering these properties – in particular, datasets with sizes useful for applying deep learning. We propose an automatic workflow to add relevant annotations to these datasets, performing manual annotations where required. In particular, the contributions of this paper are:

- We propose the People@Places dataset, based on Places365, adding bustle (6 classes) and shot type (9 classes) annotations.
- We propose the ToDY (time of day/year) dataset, based on Skyfinder [17], adding time of day (5 classes) and season (4 classes) annotations.
- We provide a baseline for the classification tasks on these datasets, using an efficient state of the art approach.
- We provide the toolchains that were used to create the two datasets, which can be used to replicate this approach for other datasets.

The rest of this paper is organized as follows. After discussing related work in Section 2, Section 3 describes the dataset creation process for People@Places, and Section 4 describes the creation process for ToDY. We present experimental results using the baseline in Section 5 and Section 6 concludes the paper.

2 Related work

We review related work on location, shot type and time of day/season classification, and for the detectors used for automatic dataset annotation. To the best of our knowledge, there is no existing work on bustle classification. The closest tasks seem to be people counting or crowd estimation, but those differ as we consider both persons and vehicles, while we are not interested in the exact numbers.

For *location type classification*, many traditional classification architectures, such as the VGG or ResNet families have been applied. Global covariance pooling is proposed in [26] to capture richer features and improve generalization. One variant of this approach, iterative matrix square root normalized covariance pooling network (iSQRT-COV-Net) used to be the best performing method on Places365, while RS-VGG16 [18] is a recent method proposing a compact model derived from VGG16. In the last few months, vision transformer models such as ViT have taken the lead [28]. A recent extension using large transformer models (86-632M parameters), and self-supervision using masked autoencoders (MAE) is to the best of our knowledge currently the best performing model for classification on Places365.

Like other computer vision tasks, *shot type* (sometimes referred to as *shot size*) *classification* is primarily addressed with deep learning approaches, either approached using CNNs directly for classification [20], using general semantic segmentation [4] or focusing on separating the subject from the background and feeding the regions into a two-stream network [19]. One issue with shot type classification is that the datasets

used in many works are not accessible, as they rely on materials from motion picture films that cannot be distributed due to copyright restrictions.

The *classification of time of day* and *season* is a topic that seems to be somewhat neglected. An early work, [7] proposes a system for season classification, but relies on color histograms and the amount of exposed skin of the depicted persons rather than on training samples. The TRECVID semantic indexing task [3] included daytime/nighttime as concepts, and the task was addressed both with traditional machine learning as well as early deep learning methods. However, except for the limitation to only two classes, the resolution and quality of this dataset is quite limited. The Youtube-8M dataset [1] covers some of the relevant classes (sunset, sunrise, night, autumn and winter), while the rest of the times of day and seasons are missing. Some vocabularies from the broadcast domain cover time of day (e.g., EBU LocationTime [1]) or season (e.g., TV-Anytime Weather [9]), but no annotated images are provided in this context.

For annotating the dataset for bustle and shot type with vehicles, persons and size of the (partial) persons in the image, we employ object detection, face detection and human pose detection. We employ YoloV4-CSP [25], which combines the CSP-Net proposed in YoloV4 [5] with an efficient model scaling strategy [25], a combination which provides us a highly accurate detector with a low inference time. RetinaFace [8] was chosen as one of the top performing methods on the challenging WIDER Face [29] hard split. For human pose detection, we employ the ROMP algorithm [22]. We chose this method because it is one of the top performing methods on a very realistic (and consequently difficult) dataset named 3D Poses in the Wild [16]. Furthermore, in contrast to other methods (like [14]) which performs also quite well on this dataset, it is a computationally efficient single-stage method which does the pose detection for all persons occurring in the image simultaneously.

One could argue that recent advances in foundation models including both vision and language such as Florence [30] or Flava [21] will sooner or later eliminate the need of training classifiers for particular task, and instead allow adaptation of models using few or no samples. This may be true, however, we strongly believe that this does not eliminate the need for specific datasets that allow the evaluation of such generic models on these tasks.

3 People@Places: Dataset for bustle and shot type classification

We amend the Places365-Standard dataset (high resolution images) with per image annotations for bustle and shot type. For bustle, we define six classes from entirely unpopulated to populated, resulting from discussions with domain experts from media production and archiving. The classification treats few large persons or vehicles separately, in order to address cases where those are in the focus of the image. Otherwise the classes use a combination of the number and size of objects, expressed by the image area covered together by these objects (see Table 1).

[1] `https://www.ebu.ch/metadata/ontologies/ebucore/ebucore_LocationTimeType.html`

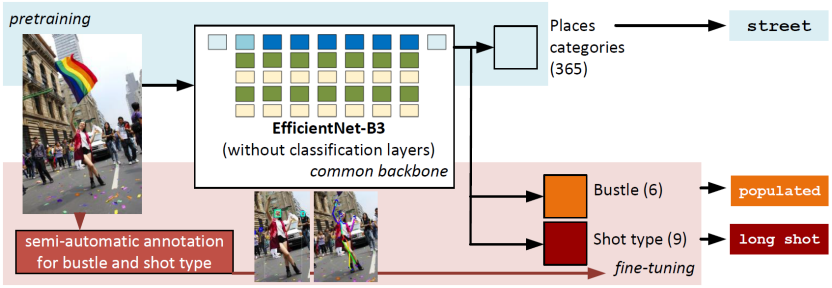

Fig. 1. Pretrained on fine-grained places categories, the backbone of the network is used to train classification heads for supercategories, bustle and shot type. Bustle/shot type annotations are created automatically (manually corrected for the validation set).

For shot types, there are a number of taxonomies that differ in the level of detail. All of them use the size of the main person depicted in the shot as reference. We use the IPTC NewsCodes scene types[2], and the lists proposed by Arijon [2], Galvane [11] and Rao et al. [19] as sources, but decided to go for a finer classification (see Table 1). As the annotations of the Places365 test split are not provided (as part of a benchmark) we work with the training and validation splits in this paper, to which we have full access.

Class	**Definition**
Bustle	
unpopulated	no persons or vehicles
few people	$<$ 3 persons, no vehicles, area $<$ 10%
few vehicles	$<$ 3 vehicles, no persons, area $<$ 20%
few large	$<$ 3 people/vehicles, any area
medium	$<$ 11 people/vehicles, area $<$ 30%
populated	more people/vehicles or covering larger area
Shot type	
extreme close-up	detail of face
close-up	head
medium close-up	cut under chest
tight medium shot	cut under waist
medium shot	cut under crotch
medium full shot	cut under knee
full shot	person fully visible
long shot	person 1/3 of frame height
extreme long shot	person <1/3 of frame height

Table 1. Definition of bustle and shot type classes.

[2] https://cv.iptc.org/newscodes/scene/

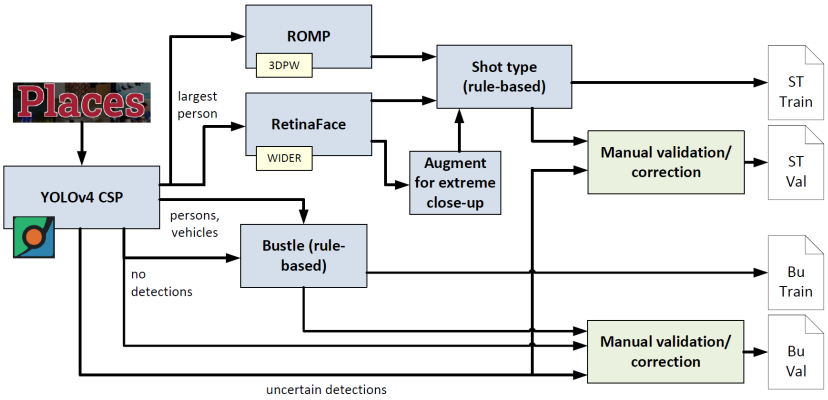

Fig. 2. Dataset creation process for bustle and shot type annotations.

The dataset creation process is semi-automatic, where automatic annotation is performed for the entire dataset, and manual verification is performed for the validation split. The process for creating the annotations is shown in Figure 2. The bustle classes depend on the presence of persons and vehicles, thus object detections for these classes are used. While person detections give a coarse indication about the size of depicted person, it is not clear which part of the person is visible. Human pose estimation and face detection are used to complement this information. In detail the process consists of the following steps.

Object detection. We run YOLOv4 CSP [25] (trained on MS COCO) over all the images, considering all detections with a score ≤ 0.1 as no occurrence. From the remaining detections, those with a score ≥ 0.5 are kept as reliable. Detections between these thresholds are considered uncertain, and the images are excluded. From the detections, persons and vehicles (i.e., the classes bicycle, car, motorcycle, airplane, bus, train, truck, boat) are kept. Based on the criteria defined in Table 1, the bustle annotation is created. In addition, the tallest person is selected and output as annotation.

Face detection. Face detection is performed using RetinaFace [8], with a model trained on WIDER Face [29], on all images that contain person detections. Multiple faces may overlap the tallest person, and it is not always straight forward to identify the correct one. We keep the face region with the largest size of the intersecting area, weighted by the detection confidence, i.e. $sc_f = (F \cap P)c_f$, where F is the face region, P is the person region and c_f is the confidence reported by the face detector.

Human pose detection. We use the ROMP [22] human pose detector (trained on 3DPW [16]), applied to a cropped out image of the tallest detected person (resp. the visible part of it). We obtain a 2D skeleton (SMPL [15] with 54 points), of which we use 10 (pelvis, left/right foot, head, left/right hip, thorax, left/right knee, spine).

Person size estimation. In order to filter unreliable detections, we filter pose and face detections for which $\max(w_D,h_D) \geq \tau \min(w_P,h_P)$, where w and h denote width and height, D denotes the pose/face detection bounding box and P denotes the person detection bounding box. τ is set to 0.1 for faces, and 0.6 for poses. If a reliable pose is found, we use it for person size estimation. We use the legs only if they appear to be stretched, i.e. head and at least one foot are on different sides of a horizontal line through the pelvis point, and the hip to feet distance is larger than the thorax to pelvis distance. If the legs are used, we check if feet and hip are on different sides of the knee (at least for one leg), otherwise we ignore the feet. If head to feet is visible, this determines the person size, otherwise we estimate the size of the part of the body not considered reliable to get the overall size measurement. We use ratios of body proportions from [10], a compact visualisation can be found on Wikipedia[3]. This is also done if only the face detection is usable. If neither pose nor face are available, we use the person detection to determine long and extreme long shots from the person height, if the person bounding box does not extend to the lower image border.

Augmentation for extreme close-up. As we found that extreme close-ups are rare in the dataset, we augment it by sampling cropped images from all close-up shots. If the larger side of a face bounding box is at least $s_{\min}$ pixels, we determine a randomly sized bounding box with $w \in [s_{\min}, 0.75 w_D]$ and $h \in [s_{\min}, 0.75 h_D]$, with $s_{\min}=175$.

Verification (validation split only.) For verification, we import the set of images into the CVAT annotation tool[4]. Each image's bustle and shot type annotation is initialized from the automatic annotation. A single annotator reviewed and corrected around 1,300 images. The accuracy of the automatically created annotations against the manually checked ones is provided in Table 4.

Data sampling. From the training set we randomly sample 100K images per class, for validation we sample 100 images per class from the manually corrected set (images used for the bustle and shot type tasks may partly overlap which is not an issue since they are treated as independent classification problems).

The annotations for bustle and shot type as well as the code of the toolchain used to create it are provided at `https://github.com/wbailer/PeopleAtPlaces`.

4 ToDY: Dataset for time of day and season

In order to build a dataset, we need a large scale outdoor dataset. We amend the Skyfinder dataset, which is a subset of the Archive of Many Outdoor Scenes (AMOS) dataset [17] dataset, consisting of about 1,500 weather webcam images per camera from 53 webcams, each covering one or multiple years. The images come with location (see Figure 3 left for a plot), date and time metadata, image timestamps (in UTC), basic weather conditions and a number of derived attributes. We aim to label each of the images with time of day and season based on the available metadata. The time

[3] `https://en.wikipedia.org/wiki/Drawing`
[4] `https://github.com/openvinotoolkit/cvat`

of day classes and their definitions are listed in Table 2, the season classes are the meteorological seasons [24], i.e., spring, summer, fall and winter.

As the location of the webcams from which the images were collected are known, as well as the dates and times when the images were taken, we can derive the season from the date and the hemisphere, and we can determine the time of the day based on the sun's position. We calculate the sun's elevation over/under the horizon at the location and time of the image, using the PyEphem[5] library. Note that this calculation will assume a horizon in a flat landscape, not considering any mountains or buildings. We are aware of this limitation, but still assume that the calculated position will be a useful approximation of the real situation.

There are multiple definitions of dusk and dawn, and we use the one for civil dusk/dawn [6], which defines begin of dusk/end of dawn when the sun is 6° below the horizon. While the begin of sunrise/end of sunset is clearly defined with the upper tip of the sun disk being just/still visible, there is not such a clear definition of the end of sunrise/begin of sunset. As the visual effect of sunrise/sunset extends beyond the point where the sun is fully visible, we chose to set this mark at the sun being 3° above the horizon. A visualization of those definitions is shown in Figure 3 (right). In addition, it needs to be considered whether a location is sufficiently far north/south, so that polar night or day occur, and thus no sunset/sunrise happens.

Based on this information, we derive season and time of day images for each image in the dataset. However, we observe three main issues with the data: (i) noisy images, in particular during nighttime, (ii) incomplete images (due to data loss when transmitting the image from the camera) and (iii) inaccurate timestamps. In order to estimate the noise level, we use the mask for the sky region provided for the Skyfinder dataset, as the sky region does hardly contain structures with strong gradients. We split the image into 8×8 patches, and we calculate the standard deviation of all patches containing at least 80% sky, and determine the noise level as the median of the standard deviations in these patches. In order to handle incomplete images, we calculate a RGB histogram of the image, and remove all images where one value covers more than 50% of the pixels of the image.

The time provided in the metadata should match the time stamp of the downloaded image file, when corrected by the UTC offset. However, even with a tolerance of 15 minutes, this does not hold for about 2/3 of the images. This is in particular a problem for classifying twilight, sunset and sunrise, as this inaccuracy may change the correct class. As we cannot tell which of the two times is correct, we decided to manually check the images. We import the set of images into the CVAT annotation tool[6], and initialize the time of day with the automatically determined value. About 10K images have been manually checked and the annotations have been corrected when necessary.

The toolchain also supports augmentation of the data by cropping versions of the images with a smaller portion of sky region. From the sky annotations of the dataset, a horizon line is determined as the 0.9 quantile of lowest sky pixels in each column. Then images with the same aspect ratio as the original image but different fractions

[5] `https://rhodesmill.org/pyephem/`
[6] `https://github.com/openvinotoolkit/cvat`

of the height above this horizon line are sampled. As the annotations are global, they are still valid for the modified images.

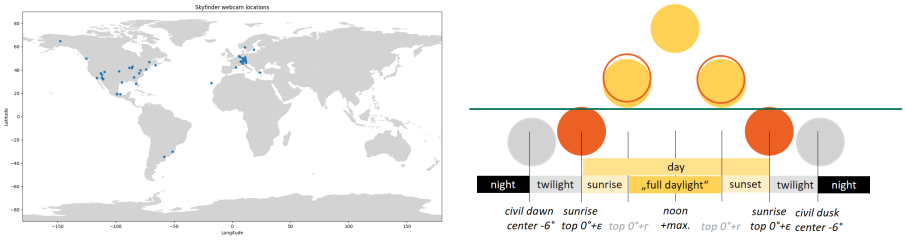

Fig. 3. Location of webcams in the Skyfinder dataset (left), visualization of the times of day used (right).

Class	Definition
night	night time
twilight	before sunrise/after sunset, using the definition of civil twilight
sunrise	sun above horizon, until fully above horizon
sunset	sun above horizon, after being fully above horizon
fulldaylight	sun completely above horizon
day	day time, i.e. fulldaylight, sunrise or sunset
	(not used as a separate class, can be derived from the other classes)

Table 2. Definition of time of day classes.

We split the resulting season and time of day annotations into balanced training and test sets. This results in 2,790 training files and 311 validation files per class for season, and 986 training files and 110 validation files per class for time of day.

The annotations for time of day and season as well as the code of the toolchain used to create it are provided at `https://github.com/wbailer/ToDY`.

5 Experiments

5.1 Baseline

We use EfficientNet-B3 [23] as the baseline model for location type classification and as a common backbone for all tasks. EfficientNet is a family of DNNs that differ in terms of number of parameters and performance. According to [23], the B3 variant provides a good tradeoff, and variants with better performance will have a significantly higher number of parameters. We train the model using the Pytorch Image Models framework (TIMM) [27], with a learning rate of 0.016 for 75 epochs.

To put the results of the model in relation to the state of the art, we compare the performance of the model on the validation set of the Places365 dataset against MAE [12], iSQRT-COV-Net [26] and RS-VGG16 [18]. However, all these methods have a significantly higher number of parameters as EfficientNet-B3. Still, its performance is slightly better than that of RS-VGG16. The results are summarized in Table 3. Throughout the paper, we use accuracy at rank 1 (acc@1) as the main metric.

Method	**no. params**	**acc@1**	**acc@5**
MAE (ViT-H) [12]	632M	60.3	-
iSQRT-COV-Net [26]	>26M	56.320	86.270
RS-VGG16 [18]	19M	51.680	82.040
EfficientNet-B3	12M	51.874	82.825

Table 3. Comparison on Places365 validation (365 classes).

5.2 People@Places

Method	**bustle**	**bustle0**	**bustle1**	**shot type**	
	acc@1	**acc@1**	**acc@1**	**acc@1**	**acc±1@1**
Toolchain	81.020	95.892	95.538	56.726	70.604
E2E	66.337	84.158	81.683	50.715	67.437

Table 4. Performance for bustle and shot type. Toolchain refers to the toolchain in Section 3, E2E refers to an end-to-end trained classifier.

The results for bustle and shot type classification are provided in Table 4. We compare the results of the computationally quite demanding annotation toolchain as described in Section 3 with the classifier trained on the datasets. The models are trained for 25 epochs (50 for shot type) with a learning rate of 0.016. For bustle classification, we observe that the results obtained from the classifier are significantly worse than that obtained with the detectors in the annotation process. To investigate this further, we introduce two binary variants of the problem: *bustle0* classifies class *unpopulated* against all others, and *bustle1* classifies{*unpopulated*, *few people*, *few vehicles*} against all others. It turns out that in these cases the performance of the classifier is closer to that of the detector toolchain. Our interpretation is that the network can well discriminate the presence of people or vehicles, but responds similarly for images with different count or size of objects, which makes it more difficult to discriminate the intermediate classes. This means that if a binary bustle classification is needed, this can be done efficiently with the classifier, while for the multi-class problem, the (computationally more expensive) detector-based approach used for dataset annotation provides better results.

For shot type classification, we observe that the results come closer to that of the annotation toolchain, but still stay below. We observe that many of the wrongly classi-

fied shots are those in nearby classes (e.g., medium shot vs. medium full shot). We thus add an evaluation metric for measuring classification into the correct or an adjacent class, which we call *acc*±1*@1*. We can observe that the performance of the annotation toolchain is in this case significantly higher, and additionally the gap between the performance of the classifier and the toolchain is reduced. For practical cases in editing, shots with similar types (which may be border cases) might already be a useful result.

5.3 ToDY

Pretraining	**ToD acc@1**	**ToD+ acc@1**	**Season acc@1**
none	63.918	20.000	28.310
ImageNet	52.577	66.182	84.225
Places365	54.639	69.818	86.197

Table 5. Top-1 accuracy for time of day and season classification using EfficientNetB3. The pretraining column specifies the base model being used, ToD+ refers to the time of day annotations after manual revision.

The results for bustle and shot type classification are provided in Table 5. The models are trained for 450 epochs for season and 1,000 epochs for time of day (stopping early if a performance ceiling is reached) with a learning rate of 0.016. We compare the performance when training EfficientNet-B3 from scratch and from models pretrained on ImageNet and Places365. We provide two results for time of day: ToD refers to the automatically generated annotations, and ToD+ to the annotations after manual corrections. Overall, the performance starting from a pretrained model is better than starting from scratch, and pretraining on Places365 provides slightly better results than pretraining on ImageNet. We assume this is due to the fact that Skyfinder images are more similar to images in many categories in Places365 than to those in ImageNet. The results of 86% accuracy for season and almost 70% accuracy for time of day show that the resulting classifiers are practically usable.

There is one anomaly, which is the high score for training time of day with automatically generated annotations from scratch. However, this seems to be due to a particular initialization, which already yields this score in the initial iteration, from which it does not change significantly during training. When applying the manual corrections, the performance falls to random when training from scratch.

6 Conclusion

We have proposed two datasets to address relevant classification tasks in visual media production and archiving: one addressed bustle and shot type classification, the other season and time of day classification. We provide toolchains for generating the additional annotations, as well as the datasets, which include manually verified and corrected subsets. The datasets are useful for classifying these properties in

images, and the toolchains enable adding these annotations to other similar datasets with limited manual effort. As a baseline, we provide experimental results using EfficientNet-B3 for the four tasks on the two datasets.

Acknowledgments. The research leading to these results has been funded partially by the program "ICT of the Future" by the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) in the project "TailoredMedia" and from the European Union's Horizon 2020 research and innovation programme, under grant agreement n° 951911 AI4Media (`https://ai4media.eu`). The authors would like to thank Martin Winter, Hermann Fürntratt and Stefanie Onsori-Wechtitsch for support with the face detector and annotation tool setup, and Levi Herrich for checking and correcting the time of day annotations.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Arijon, D.: Grammar of the film language. Silman-James Press (1991)
3. Awad, G., Snoek, C.G., Smeaton, A.F., Quénot, G.: Trecvid semantic indexing of video: A 6-year retrospective. ITE Transactions on Media Technology and Applications **4**(3), 187–208 (2016)
4. Bak, H.Y., Park, S.B.: Comparative study of movie shot classification based on semantic segmentation. Applied Sciences **10**(10), 3390 (2020)
5. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. ArXiv **abs/2004.10934** (2020)
6. Boggs, S.: Seasonal variations in daylight, twilight, and darkness. Geographical Review **21**(4), 656–659 (1931)
7. Cheng, P., Zhou, J.: Automatic season classification of outdoor photos. In: 2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics. vol. 1, pp. 46–49. IEEE (2011)
8. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641 (2019)
9. ETSI: Ts 102 822-3-1 v1.9.1 - broadcast and on-line services: Search, select, and rightful use of content on personal storage systems (tv-anytime); part 3: Metadata; sub-part 1: Phase 1 - metadata schemas. Tech. rep. (2015)
10. Fairbanks, A.T., Fairbanks, E.F.: Human proportions for artists. Fairbanks Art and Books (2005)
11. Galvane, Q.: Automatic Cinematography and Editing in Virtual Environments. Ph.D. thesis, Université Grenoble Alpes (ComUE) (2015)
12. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
13. Huurnink, B., Hollink, L., Van Den Heuvel, W., De Rijke, M.: Search behavior of media professionals at an audiovisual archive: A transaction log analysis. Journal of the American society for information science and technology **61**(6), 1180–1197 (2010)
14. Kissos, I., Fritz, L., Goldman, M., Meir, O., Oks, E., Kliger, M.: Beyond weak perspective for monocular 3d human pose estimation. In: European Conference on Computer Vision (ECCV). pp. 541–554 (01 2020)

15. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
16. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV) (sep 2018)
17. Mihail, R.P., Workman, S., Bessinger, Z., Jacobs, N.: Sky segmentation in the wild: An empirical study. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–6 (2016). https://doi.org/10.1109/WACV.2016.7477637, acceptance rate: 42.3%
18. Qassim, H., Verma, A., Feinzimer, D.: Compressed residual-vgg16 cnn model for big data places image recognition. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). pp. 169–175. IEEE (2018)
19. Rao, A., Wang, J., Xu, L., Jiang, X., Huang, Q., Zhou, B., Lin, D.: A unified framework for shot type classification based on subject centric lens. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 17–34. Springer (2020)
20. Savardi, M., Signoroni, A., Migliorati, P., Benini, S.: Shot scale analysis in movies by convolutional neural networks. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 2620–2624. IEEE (2018)
21. Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., Kiela, D.: Flava: A foundational language and vision alignment model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15638–15650 (2022)
22. Sun, Y., Bao, Q., Liu, W., Fu, Y., Black, M.J., Mei, T.: Monocular, one-stage, regression of multiple 3d people. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11179–11188 (2021)
23. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)
24. Trenberth, K.E.: What are the seasons? Bulletin of the American Meteorological Society **64**(11), 1276–1282 (1983)
25. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Scaled-yolov4: Scaling cross stage partial network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13029–13038 (2021)
26. Wang, Q., Xie, J., Zuo, W., Zhang, L., Li, P.: Deep cnns meet global covariance pooling: Better representation and generalization. IEEE transactions on pattern analysis and machine intelligence (2020)
27. Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). https://doi.org/10.5281/zenodo.4414861
28. Xiao, T., Dollar, P., Singh, M., Mintun, E., Darrell, T., Girshick, R.: Early convolutions help transformers see better. Advances in Neural Information Processing Systems **34** (2021)
29. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5525–5533 (2016)
30. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)
31. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)