Supplementary Material - Temporal Dynamics in Visual Data: Analyzing the Impact of Time on Classification Accuracy

Tom Pégeot* Eva Feillet*,+ Adrian Popescu* Inna Kucher* Bertrand Delezoide†

(*) Université Paris-Saclay, CEA, LIST
F-91120 Palaiseau, France
name.surname@cea.fr

(+) Université Paris-Saclay, CentraleSupélec, MICS
F-91190 Gif-sur-Yvette, France
name.surname@centralesupelec.fr

(†) Amanda
F-75008, Paris, France

A. Annotation interface

In Figure 1 we show the interface used by annotators to label the images of VCT-107 . In this example, the annotator checks the annotation of images uploaded in 2008 and retrieved from Flickr using the keyword *carrot*. A class definition is provided to the annotator on top of the images. The images are grouped by cluster to facilitate the annotation. They are also sorted by cosine similarity to their cluster's center for faster annotation (from left to right: highest similarity to lowest similarity). In this example, the images from the first and fourth rows are all validated by the annotator (green boxes). The annotator can simply double-click on the last image of a row (the image with the lowest cosine similarity) to validate all the images of the row. In Figure 2, we show an example of a cluster whose images are not all validated by the annotator. Here, the annotator selected only a subset of the images. Finally, the interface returns a JSON file with the validated images for each annotator.

B. VCT-107 Content

A Box plots showing the number of images per data collection period and VCT-107 class is given in Figure 3.

In Table 1, we provide the list of classes corresponding to each metaclass. As explained in Section 3, each metaclass corresponds to a topic used to prompt ChatGPT-4o for obtaining related class names. In Tables 3, 4 and 5 we provide the total number of samples collected for each class and each period in VCT-107 .

C. Implementation details

All our experiments are implemented using PyTorch.

C.1. Model checkoints

Our experiments use the following pre-trained models:

- ResNet18 pre-trained on ILSVRC, available at https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html
- ViT-B/14 pre-trained on the LVD-142m dataset with DINOv2, available at https://dl.fbaipublicfiles.com/dinov2/dinov2_vitb14/dinov2_vitb14_pretrain.pth,
- ViT-B/16 pre-trained on ImageNet-21k, available at https://huggingface.co/timm/vit_base_patch16_224.augreg_in21k.
- ViT-B/16 pre-trained on ILSVRC, available at https://pytorch.org/vision/main/models/generated/torchvision.models.vit_b_16.html.
- ViT-B-16 and ViT-L-14 based CLIP with openAI pretraining, available at https://github.com/mlfoundations/open_clip

C.2. DIL experiments

In the following, we report the hyperparameter choices made in the DIL experiments from Subsection 4.4.

The nearest class mean classifier (NCM) relies on a frozen encoder. For each class, it computes a prototype by averaging the embedding vectors of the training samples belonging to this class. We implement NCM using the cosine distance.

Our implementation of FeCAM [2] is based on the original repository of the authors[1]. Results are obtained using the following hyperparameters for covariance shrinkage.

- FeCAM with a single covariance matrix: $\alpha_1 = 1.0, \alpha_2 = 0.0$

[1] https://github.com/dipamgoswami/FeCAM

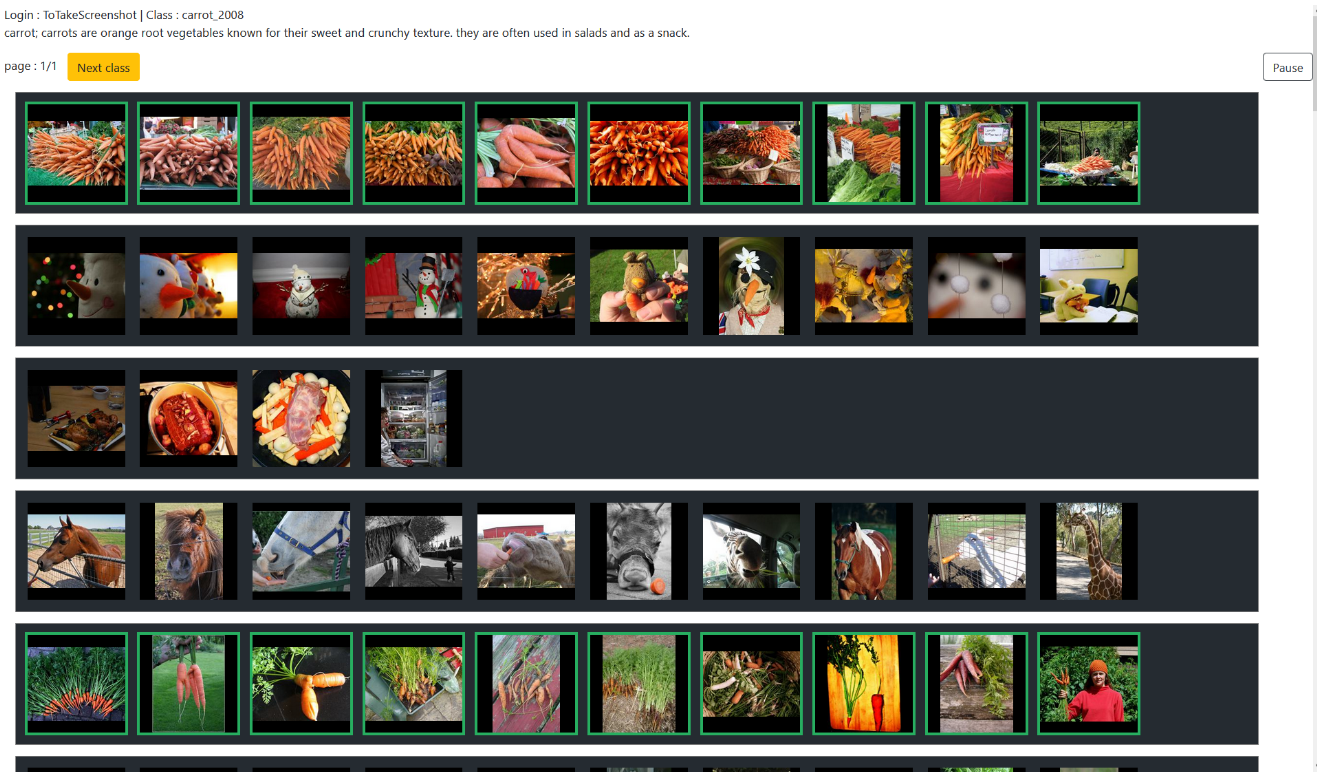

Figure 1. Annotation interface for the class “carrot”. The images are grouped by cluster. An annotator checks the assigned label manually.

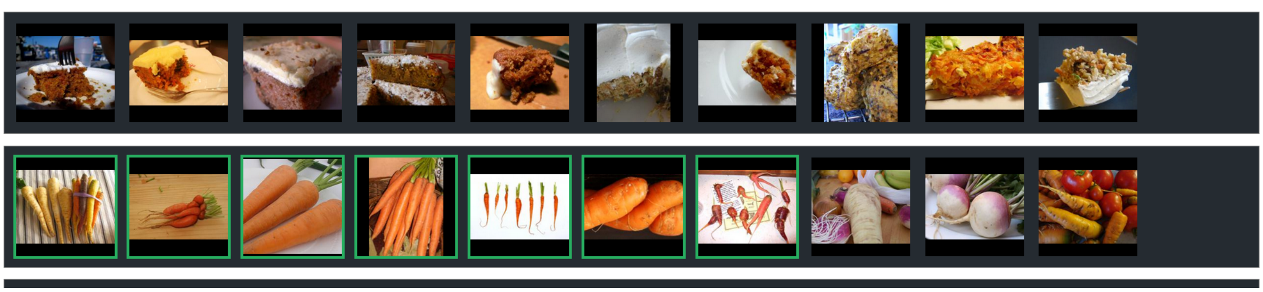

Figure 2. Annotation interface for the class “carrot”. Example of a cluster containing images that are only partially accepted as examples for the class “carrot” by the annotator.

- FeCAM with one covariance matrix per class: $\alpha_1 = 10.0, \alpha_2 = 1.0$

Our implementation of RanPAC [3] is based on the original repository of the authors[2]. We keep the hyperparameters unchanged, notably $M = 10,000$ the new embedding size for projecting the features.

The cumulative strategies “replay-20” and “accumulate” store either 20 or 200 training images per class and per period. At each step of the incremental process, we train a linear layer on top of the pre-trained encoder (linear probing). The linear layer is trained for 20 epochs using the SGD optimizer with a momentum set to 0.9 and a weight decay set to $4e^{-5}$, and a cosine learning rate scheduler with a starting value of 0.1.

[2] https://github.com/RanPAC/RanPAC

D. Zero-shot classification with CLIP

In section 4.2 results are provided for CLIP ViT-B/16 and CLIP ViT-L/14 using linear probing. In the following, we provide the classification accuracy of the same ViT-B/16 network but in a zero-shot setting. These results provide insight into the relative difficulty of each period, which may vary slightly.

To make this classification we used the label of each class as input for the textual encoder. Textual class labels

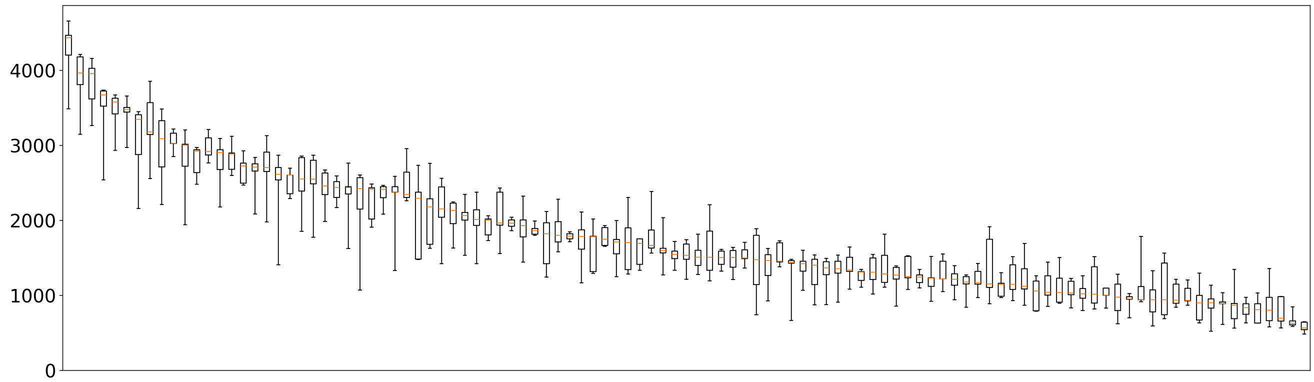

Figure 3. Box plots showing the number of images per data collection period and VCT-107 class. Class labels are provided in the appendix.

Metaclass (topic)	#classes	Class names
Household Objects	7	dining table, mug, chair, sofa, pillow, stove, spoon
Animals	31	dog, cat, horse, snake, fish, parakeet, frog, duck, giraffe, turtle, elephant, rabbit, wallaby, hippopotamus, prairie dog, lemur, meerkat, salamander, iguana, zebra, african penguin, crocodile, donkey, chimpanzee, lion, gorilla, leopard, gibbon, toucan, polar bear, sloth
Sporting Equipment	7	bicycle, basketball, kite, surfboard, skateboard, snowboard, soccer ball
Plants	25	mushroom, rose, hydrangea, poppy, dahlia, orchid, peony, crocus, sunflower, hibiscus, columbine, tulip, amaryllis, lavender, tomato, cosmos, pansy, lilac, iris, foxglove, hyacinth, daffodil, strawberry, broccoli, artichoke
Apparel	8	suit, sneakers, dress, scarf, raincoat, t-shirt, tie, hoodie
Food	8	pasta, ramen, cupcakes, pancakes, croissant, sushi, ice cream, burger
Vehicles	11	car, airplane, sailboat, motorcycle, bus, tram, truck, canoe, helicopter, tuk-tuk, yacht
Electronic Devices	2	laptop, headphones
Buildings	8	church, skyscraper, house, windmill, greenhouse, gas station, restaurant, observation tower

Table 1. List of the metaclasses (topics) and classes of the VCT-107 dataset. The classes of a given metaclass are ordered by decreasing median number of images per period. See Tables 3 to 5 for the detailed number of samples per class.

2007-08	2010-11	2013-14	2016-17	2019-20
86.6%	87.4%	86.7%	85.3%	84.0%

Table 2. Zero-shot accuracy of ViT-B/16 CLIP model on each period

are listed in Table 1. The results, presented in Table 2, indicate that the final period (2019-2020) is slightly more challenging than the others.

E. Data augmentation

In Section 4, we apply standard data augmentation techniques. Below, we present the results of preliminary tests that guided our choice of these specific augmentations.

To do so, we used ResNet18, as it is the easiest model to train from scratch with a "small" dataset such as VCT-107. We selected three sets of data augmentations. (i) The first does not include any data augmentation. (ii) The second uses the most common data augmentation operation, which consists of randomly cropping the images and then flipping them horizontally with a probability of 0.5. (iii) Finally, the third set of data augmentations includes additional transformations, such as random adjustments to luminosity, saturation, contrast, and hue, randomly rotating the images, random cropping, and finally randomly flipping the images horizontally. The factors for luminosity, saturation, and contrast are picked from the range [0.6, 1.4], the hue factor is chosen from the range [-0.4, 0.4], and the rotation is uniformly selected from the range [-20°, +20°].

In Figure 5, we can see that the model is affected by the change in the data collection period, regardless of the data

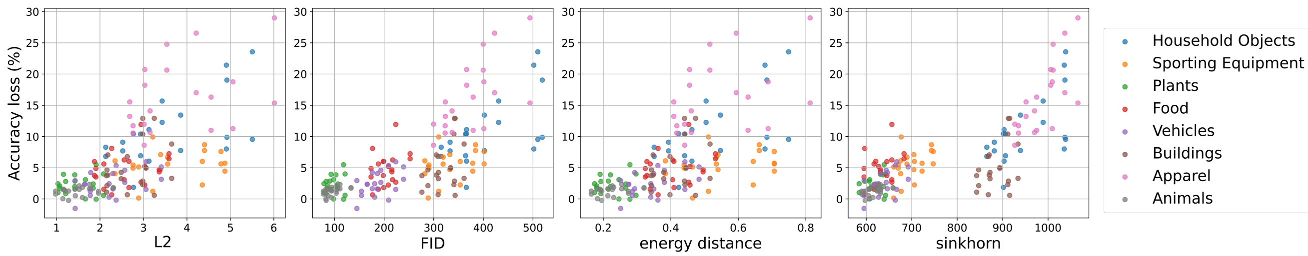

Figure 4. Relative accuracy loss over time for the classes of the general VCT-107 topics as a function of distribution shift measured with four metrics. In this figure, all the samples belonging to the *metaclass* are considered as samples from a single distribution. This differs from the results given in the Section 5.2 for which a distribution only contains one class

augmentations chosen. However, we note that not performing any data augmentation generally reduces the model's accuracy.

In Figure 6, we observe that in the case of linear probing with a pre-trained model, applying either more data augmentation (option (iii)) or no data augmentation at all (option (i)) leads to worse performance. This can be explained by the fact that the feature extractor was pre-trained using only the data augmentations corresponding to the intermediate data augmentation set (option (ii)).

In conclusion of these experiments, we decided to only use the standard data augmentations that correspond to those used in the pre-training of the backbones. We also maintain these values for tests with non-pre-trained networks because the results from Figure 1 show that we do not gain any improvement by using additional data augmentations.

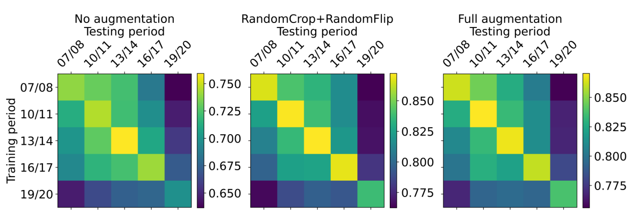

Figure 5. Accuracy when training a ResNet18 from scratch on one period and testing on the others. The experiment was done with three sets of data augmentation.

F. Shift measured on the entire topic

In Section 5, for each class, we measured the distance between the distribution of each period. Here, we use the same distances but consider the distributions at the topic level instead of the class level. Intuitively, the distribution will have a greater chance of being multimodal. This can be important for FID as it supposes multivariate normality [1]. Therefore, this result is only for informational purposes.

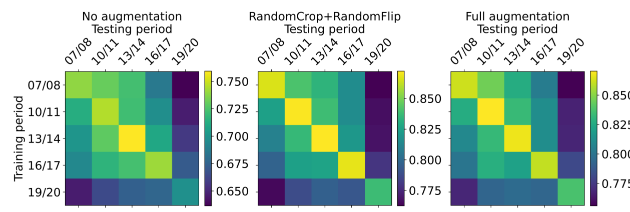

Figure 6. Accuracy when training a linear probes on a ResNet18 on one period and testing on the others. The pretraining was done with ILSVRC. The experiment was done with three sets of data augmentation.

In Figure 4, we can see that in this case, FID fails to assign smaller values to *Sporting Equipment* than *Household Objects*. This would have been the expected result as the metaclass *Sporting equipment* suffers from less loss in accuracy when generalizing to other periods.

Meanwhile, the results for Sinkhorn stay very similar to those observed in Section 5.

References

[1] D.C. Dowson and B.V. Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982. 4

[2] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[3] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2

Class names $A \rightarrow H$	2007-2008	2010-2011	2013-2014	2016-2017	2019-2020
african penguin	1367	1491	1274	1449	877
airplane	3625	3578	3419	3669	2933
amaryllis	1788	1822	1846	1714	1753
artichoke	888	809	1032	631	629
basketball	1210	1503	1640	1597	1374
bicycle	3205	3001	3015	2722	1940
broccoli	1281	1148	976	798	621
burger	1001	1297	900	670	633
bus	2305	2516	2592	2435	2169
canoe	1930	2321	2005	1779	1441
car	3736	3722	3522	3669	2538
cat	4179	3965	4211	3807	3147
chair	2003	2345	2065	2106	1532
chimpanzee	1453	1221	1551	1220	1049
church	3619	3956	4158	4025	3263
columbine	1892	1990	1815	1864	1800
cosmos	2384	1870	1664	1631	1564
crocodile	1644	1318	1508	1336	1082
crocus	2132	2228	2247	1956	1631
croissant	942	1327	1072	591	779
cupcakes	1915	1748	1151	1105	889
daffodil	1253	1079	1518	1234	1528
dahlia	2869	3100	2916	3211	2765
dining table	2603	2421	2150	2568	1070
dog	4464	4655	4433	4200	3485
donkey	1814	1283	1534	1107	1173
dress	1965	2374	2431	1934	1558
duck	2081	2301	2447	2466	2412
elephant	2374	1931	2010	2141	1421
fish	2868	2704	2611	2540	1407
foxglove	1018	1544	1210	1499	1307
frog	2341	2456	2671	2631	1985
gas station	1244	1273	1345	1170	1098
gibbon	1504	1037	910	1228	894
giraffe	2586	2449	2377	2375	1330
gorilla	971	987	1162	1303	1147
greenhouse	1481	1463	1436	1429	665
headphones	1355	1454	1535	1296	910
helicopter	971	1150	1169	1425	1320
hibiscus	1960	2004	2040	1919	1863
hippopotamus	1997	1553	1708	1744	1248
hoodie	848	587	628	607	660
horse	3569	3144	3177	3852	2555
house	2550	2854	2834	2390	1852
hyacinth	1392	1216	1372	1268	856
hydrangea	2851	3217	3023	3025	3161

Table 3. Number of samples per period for class names from A to H.

Class names $I \rightarrow S$	2007-2008	2010-2011	2013-2014	2016-2017	2019-2020
ice cream	1206	932	1094	926	868
iguana	1601	1323	1423	1455	1067
iris	1347	1311	1107	1315	1200
kite	1801	1888	1474	1144	743
laptop	1820	2117	1966	1421	1243
lavender	1664	1902	1928	1746	1653
lemur	1742	1481	1533	1683	1215
leopard	1078	1146	1516	1406	930
lilac	1611	1588	1503	1321	1414
lion	1152	1272	1248	1179	840
meerkat	1398	1602	1278	1508	1814
motorcycle	2657	2708	2839	2753	2084
mug	2177	2758	2287	1680	1627
mushroom	3474	3655	3443	3502	2971
observation tower	913	1034	892	888	612
orchid	2679	2597	2884	2896	3120
pancakes	1381	1515	1015	896	817
pansy	2035	1599	1273	1626	1566
parakeet	2603	2603	2353	2290	2694
pasta	2374	2294	2733	1483	1479
peony	2468	2723	2926	2762	2496
pillow	1086	1691	1120	1355	867
polar bear	1096	1095	1003	1000	831
poppy	2924	2970	2940	2635	2481
prairie dog	2305	1704	1901	1283	1344
rabbit	1615	2110	1873	1784	1167
raincoat	562	869	891	1346	688
ramen	1544	1718	1589	1333	1489
restaurant	947	981	1023	701	954
rose	2875	3406	3344	3447	2158
sailboat	2903	3091	2678	2939	2177
salamander	1445	1382	1455	1700	1726
scarf	2207	1856	1508	1335	1194
skateboard	1395	1134	1287	1215	941
skyscraper	2704	3129	2909	2651	1978
sloth	1134	829	953	900	521
snake	3483	3327	2712	3087	2211
sneakers	2350	2435	2763	2447	1624
snowboard	1058	1262	1190	789	794
soccer ball	940	915	1118	1785	944
sofa	1467	1264	1543	1624	926
spoon	835	972	749	634	887
stove	1431	1563	941	688	743
strawberry	1262	1443	1002	1040	852
suit	2549	2487	2799	2867	1775
sunflower	2016	2003	2062	1806	1730
surfboard	1519	1234	1121	1229	922
sushi	841	1149	1213	894	935

Table 4. Number of samples per period for class names from I to S.

Class names $T \rightarrow Z$	2007-2008	2010-2011	2013-2014	2016-2017	2019-2020
t-shirt	974	580	663	800	1357
tie	982	985	656	695	565
tomato	1691	1753	1753	1413	1333
toucan	963	1090	1021	1260	797
tram	2017	2415	2484	2433	1908
truck	2303	2956	2343	2641	2259
tuk-tuk	1187	1009	1033	1225	832
tulip	1800	1982	1580	2281	1713
turtle	2444	2559	2152	2041	1423
wallaby	1787	2017	1294	1782	1315
windmill	1362	1705	1608	1489	1498
yacht	649	483	565	642	542
zebra	1540	1481	1402	1142	873

Table 5. Number of samples per period for class names from T to Z.