UNIVERSITY OF LINCOLN

LITTER - PASTE

Investigating the impact of context and style transfer on image synthesis for deep learning dashcam litter detection

OAKLEIGH WEEKES 18678532
MSC INTELLIGENT VISION
RESEARCH PROJECT

AIM

- *To investigate whether pasting litter items into dashcam images to train an object detection network increases performance.*

We are concerned with…

- Context - **Where** do we paste the litter items in the images?
- Style - How do we **integrate** pastes naturally, so that they are visually feasible?

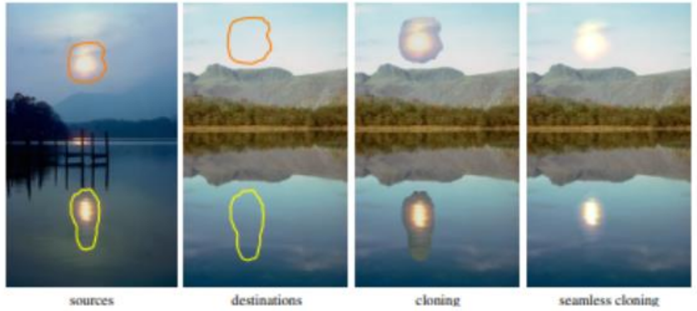
MOTIVATIONS

External Datasets

- Annotation
  - Time-consuming
  - Error-prone
- Publicly available litter datasets
- Capture stochasticity
  - Highly-varied litter
  - Highly-varied backgrounds

Composition Synthesis

- Copy-Paste can improve object detection performance (Dwibedi, Misra and Hebert, 2017)
- Could capture variety in other datasets and apply to dashcam-view context (domain adaption)
- Increases small object detection performance
  - More samples
  - Resizing strategy



Style Transfer / Blending

- NST improves performance when detecting real objects in art (Work for decompression?) (Kadish, Risi and Løvlie, 2021)
- Blending improves performance by merging light intensity gradients of a placed object with its new background (Dwibedi, Misra and Hebert, 2017)
- Visually integrates a pasted object

DATASETS & MODELS

- Source Dataset – TACO (Trash Annotations in Context) (Proença and Simões, 2020)
- Target Dataset – DashLit (LoVE) & Supervisely-Roads (External)
- Context (One option) – Segment-Anything Model (Kirillov et al., 2023)
- Style Transfer- AdaIN (Adaptive Instance Normalisation) (Gatys, Ecker and Bethge, 2015)
- Detection – YOLOv8 (Jocher, Chaurasia and Qiu,2023)

SOFTWARE USAGE

In [2]:

```
from stylecopypaste.datasets.create import dataset

#Path to directory of images/labels in yolo format
src = r"[YOLO annotated dashcam data]"

#New dataset save destination
dest = r"[output]"

#Path to training data points (csv)[filename xmin ymin ymax class width height]
points_p = r"[training points CSV]"

# AdaIN checkpoint trained on style of dashlit data

style_weights = r"[AdaIN checkpoint folder]"

#If using SAM, put original SAM trained model weights in directory also

####random placement
ds = dataset(src,dest,"random")
####points placement
ds = dataset(src,dest,"points",pts = points_p,blend=True)
####segment placement
ds = dataset(src,dest,"seg",pts = points_p)


####random placement w/style
ds = dataset(src,dest,"random",style=style_weights)

####points placement w/ style
ds = dataset(src,dest,"points",style=style_weights,pts = points_p)

####segment placement w/style.
ds = dataset(src,dest,"seg",style=style_weights,pts = points_p)


#Then assign litter to paste and run creation…

taco_p = r"[Path to TACO]"

ds.generate(taco_p)
```

ADDS synthetic frames to original supplied (DashLit) dataset

BASELINES

- DashLit
- DashLit-TACO (Simple TACO addition) -> Higher precision, lower AP & Recall.
- Precision @ 0.3 IoU threshold.

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit	0.708	0.759	0.767	0.323
DashLit-TACO	0.694	0.78	0.76	0.313

OAKLEIGH WEEKES 18678532
MSC INTELLIGENT VISION
RESEARCH PROJECT

CONTEXT EXPERIMENTS

(a) Random placement gen.

(b) Training-point placement gen.

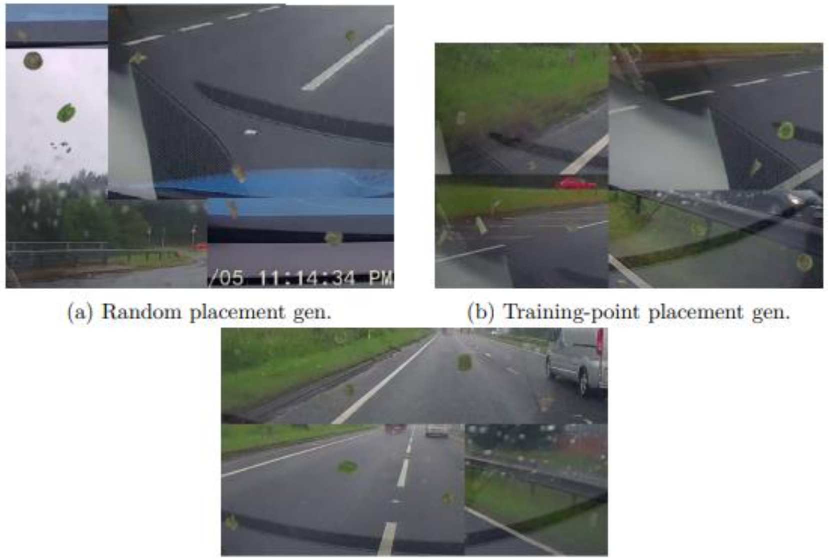
(c) ROI placement gen.

- Random
- Training-Point Placement
- Region of Interest placement (Reduced-point-seeded SAM Model) [Best against floodfill and HSV methods]
- Does not beat baseline performance, however points-based placement has best context AP.

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
Random	0.705	0.751	0.751	0.311
Points-Placement	0.698	0.757	0.759	0.317
ROI (SAM) Placement	0.703	0.748	0.752	0.31

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit	0.708	0.759	0.767	0.323
DashLit-TACO	0.694	0.78	0.76	0.313

STYLE EXPERIMENTS



(a) Random placement gen. (b) Training-point placement gen. (c) ROI placement gen.

- Added NST (trained AdaIn on TACO & DashLit samples)
- Points-based NST placement training has a significant increase in recall, with a slight reduction in AP & precision.

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
Random [NST]	0.688	0.749	0.747	0.309
Points-Placement [NST]	0.741	0.747	0.756	0.311
ROI (SAM) Placement [NST]	0.697	0.751	0.748	0.301

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit	0.708	0.759	0.767	0.323
DashLit-TACO	0.694	0.78	0.76	0.313

EXTERNAL FRAME DATA & BLURRING

- Pasted onto another road dataset and added to training, using points-based placement and Poisson blending.
- External-frame pasting with blending near-matches DashLit baseline with an increase to precision at a low threshold and AP at high thresholds.

(a) External-frame placement gen. (b) External-frame blending placement gen.

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit-ExternalGen	0.704	0.766	0.761	0.317
DashLit-ExternalGen [Blend]	0.707	0.764	0.767	0.326

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit	0.708	0.759	0.767	0.323
DashLit-TACO	0.694	0.78	0.76	0.313

FULL RESULTS

Table 5.5: Table showing all detection performance results, with the highest metrics highlighted in **bold**.

Dataset	Recall	Precision	AP @ 0.5	AP @ .5-.95
DashLit	0.708	0.759	**0.767**	0.323
DashLit-TACO	0.694	**0.78**	0.76	0.313
Random	0.705	0.751	0.751	0.311
Points-Placement	0.698	0.757	0.759	0.317
ROI (SAM) Placement	0.703	0.748	0.752	0.31
Random [NST]	0.688	0.749	0.747	0.309
Points-Placement [NST]	**0.741**	0.747	0.756	0.311
ROI (SAM) Placement [NST]	0.697	0.751	0.748	0.301
DashLit-ExternalGen	0.704	0.766	0.761	0.317
DashLit-ExternalGen [Blend]	0.707	0.764	**0.767**	**0.326**

EVALUATION/CONCLUSION

- Training-points paste placement seems to be the best method of implementing context in pastes. Neural style transfer is beneficial but requires paste context to closely match that of the litter style it was trained with.
- Using external road data with blending helps the model generalise – as adding these composed images to training does not reduce performance in any metrics significantly, but instead improves low IOU threshold precision and AP at high thresholds. This could also indicate overfitting when adding synthesised frames made with seen backgrounds (See figures).

→ Further study needs to examine using ONLY synthesised data in training.

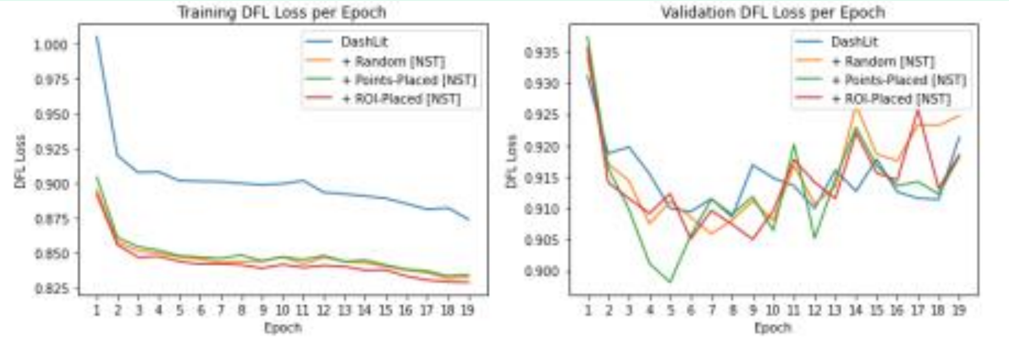


Figure 5.5: Graph showing DFL loss per epoch of training YOLOv8L on neutal style transfer generated datasets.

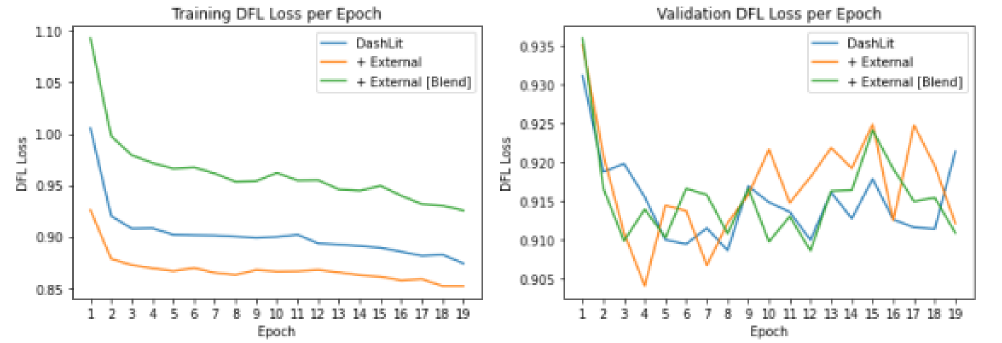


Figure 5.7: Graph showing DFL loss per epoch of training YOLOv8L on external-frame generated datasets.

BONUS: EXAMPLE 1: MCDONALDS FRIES CARTON

Original

Baseline - DashLit

Baseline – DashLit-TACO

Points-Paste-Style

External-Blend-Paste

BONUS: EXAMPLE 2: DISCARDED LICENSE PLATE

Original

Points-Paste-Style

Baseline - DashLit

External-Blend-Paste

Baseline – DashLit-TACO

BONUS: EXAMPLE 3: FLOWERS

Original

Baseline - DashLit

Baseline – DashLit-TACO

Points-Paste-Style

External-Blend-Paste

FURTHER WORK

- Examine using ONLY synthesised data in training.
- Experiment with applying blending to ALL experiments.
- Find a more accurate segmentation strategy for context (or use points more dynamically).
- Use further backgrounds and litter datasets in training – limited by storage.
- Improve memory efficiency of code solution.
- Incorporate tracking & Mapping
- Litter pick-up – robots…

REFERENCES

- Dwibedi, D., Misra, I. and Hebert, M. (2017). *Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.1708.01642.
- Kadish, D., Risi, S. and Løvlie, A.S. (2021). *Improving Object Detection in Art Images Using Only Style Transfer*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2102.06529.
- Proença, P.F. and Simões, P. (2020). *TACO: Trash Annotations in Context for Litter Detection*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2003.06975.
- Gatys, L.A., Ecker, A.S. and Bethge, M. (2015). *A Neural Algorithm of Artistic Style*. [online] arXiv.org. Available at: https://arxiv.org/abs/1508.06576.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P. and Girshick, R. (2023). Segment anything. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2304.02643.
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.