JOHANNES KEPLER
UNIVERSITY LINZ

CV Lab Project, Lab 2: Ideas

Group A0:
Pablo Díez Arrizabalaga, Christian Willdoner, René H. Reich, and Daniel Schatzl

JOHANNES KEPLER
UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Unsupervised Pixel-wise Hyperspectral Anomaly Detection via Autoencoding Adversarial Networks, Sertac Arisoy et al.

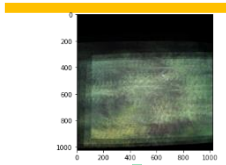
Ensemble and Random Collaborative Representation-Based Anomaly Detector for Hyperspectral Imagery, Rong Wang et al.

- Many papers on HAD (Hyperspectral Anomaly Detection)

Test images

Train images

Overlay images over camera axis

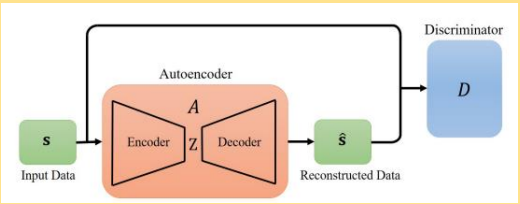


$$\delta_i = \|\boldsymbol{x}_i - \boldsymbol{X}_r \boldsymbol{a}_i\|_2.$$

(Ensemble) Random CRD for anomaly score for detecting anomaly from background

Use background pixels as training set **s** where each sample has size *mxmxL*

$$\{\hat{A}_d, \hat{D}_d\} = \arg\min_{A_d} \max_{D_d} \mathcal{L}_{Adv} + \lambda \mathcal{L}_{\mathcal{R}}$$



$$\hat{\mathbf{s}}^{(d)} = A_d(\mathbf{s}^{(d)})$$

$$\mathcal{L}_{\mathcal{R}} = \mathrm{E}_{\mathbf{s}^{(d)}}\left[\|\mathbf{s}^{(d)} - \hat{\mathbf{s}}^{(d)}\|_1\right]$$

F ϵ $R^{M1xM2xL}$ with M_1, M_2 image width, height and L channels, $F = \{f_1, f_2, ,,,, f_{M1xM2}\}$, f_i ϵ R^L

Same sample sizes as for training samples

Original RGB image assembled by feeding AE with *mxmxL* blocks of the original image,→$\hat{f}_i$ vectors are obtained → calc reconstruction error map REM

(3-D)-ANAE

WRX based detector where weights are REM values → $w_i = 1/r_i$ → low weight means anomaly

Anomaly score

$$AD(f_i) = (f_i - \hat{m})^T \hat{C}^{-1} (f_i - \hat{m})$$

Bounding Box

REM ϵ R^{M1xM2} consists of $r_i = \|f_i - \hat{f}_i\|$ for i-th pixel

Image merging/overlaying

Why?

- Expecting better results [1]

How?

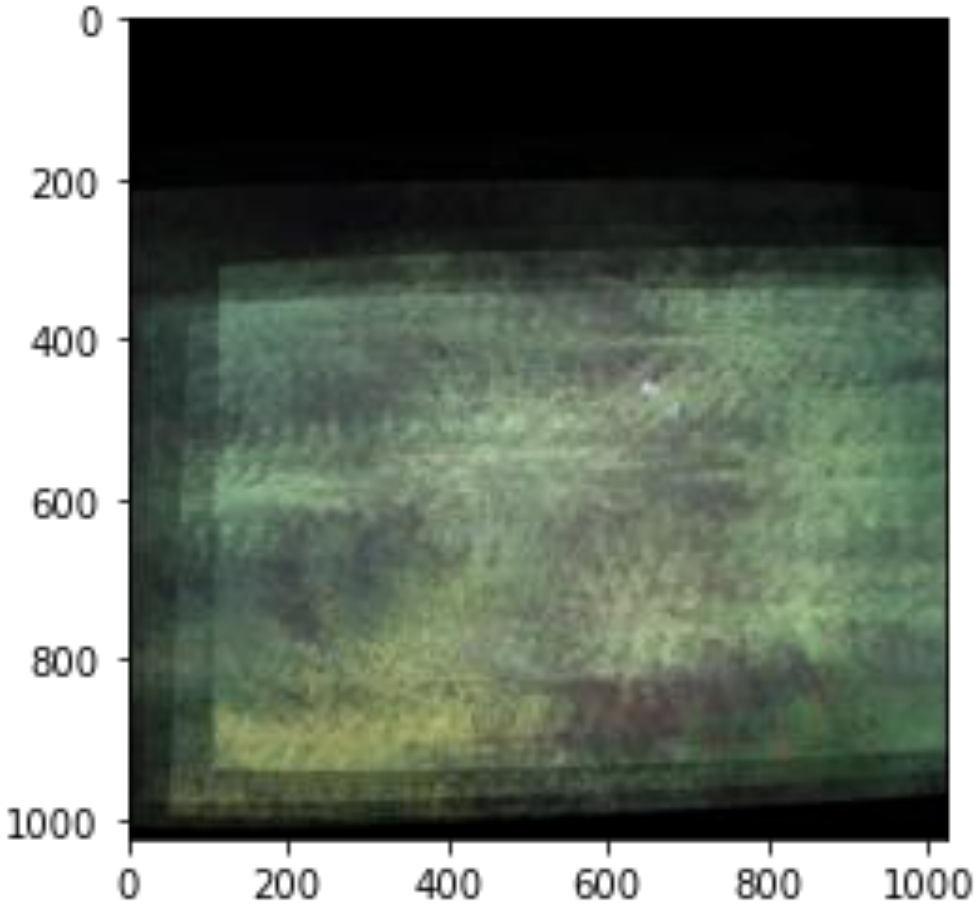
- Using homography data provided
- Center image is the base, transformed outer images are laid on top of it
- $\alpha = 0.75$

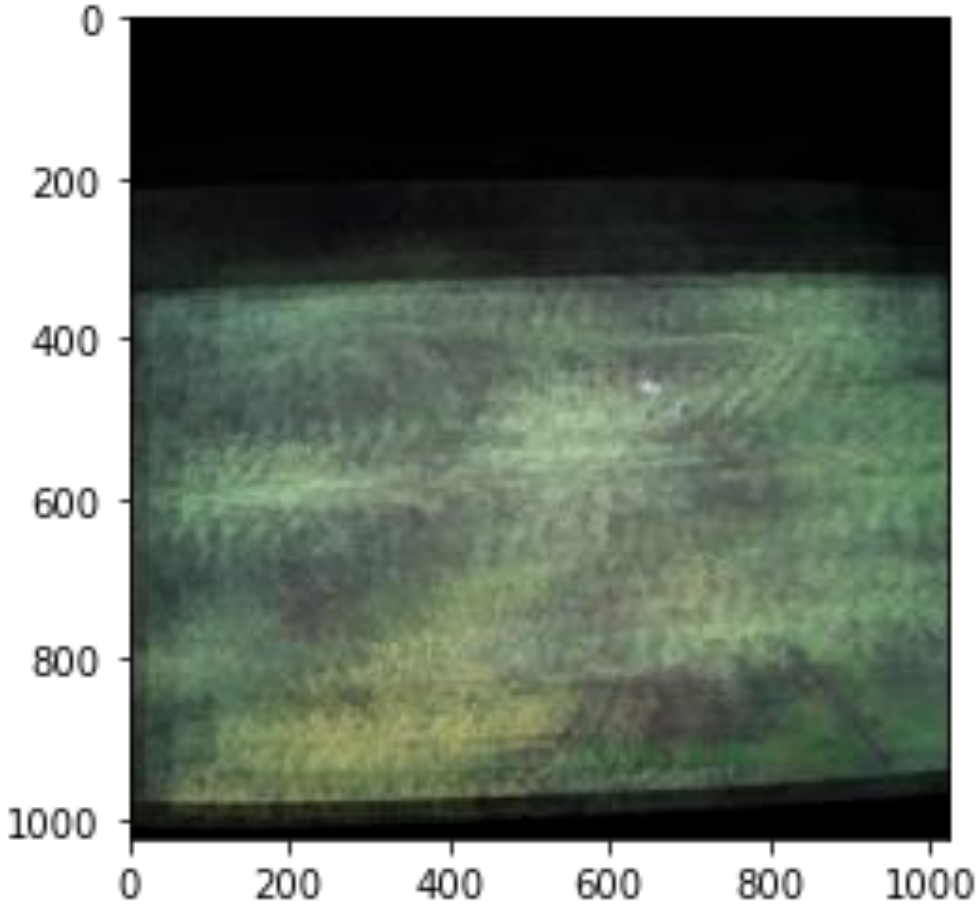
[1] Search and rescue with airborne optical sectioning [Schedl, D.; Kurmi, I.; Bimber, O.]

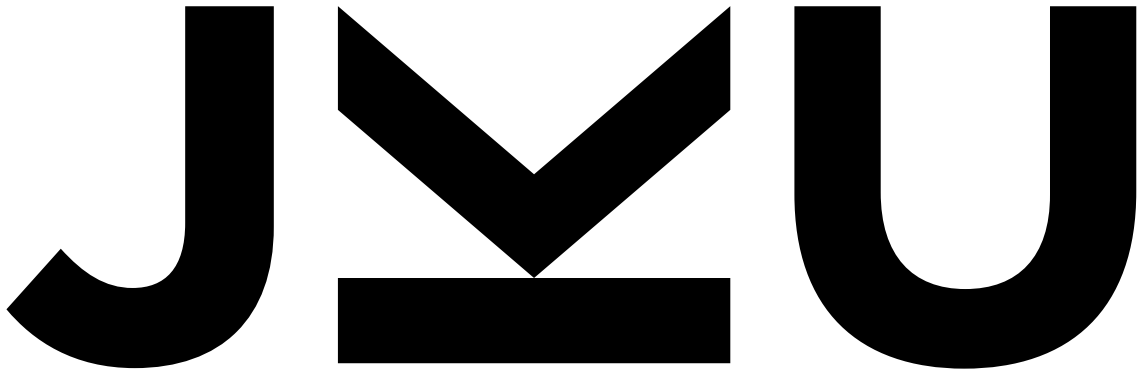
Important Python Functions

```
cv2.warpPerspective(image, homography, shape)
cv2.addWeighted(img1, alpha, img2, 1 - alpha, 0.0)
```

Results


0
200
400
600
800
1000
0
200
400
600
800
1000


0
200
400
600
800
1000
0
200
400
600
800
1000



JOHANNES KEPLER
UNIVERSITY LINZ