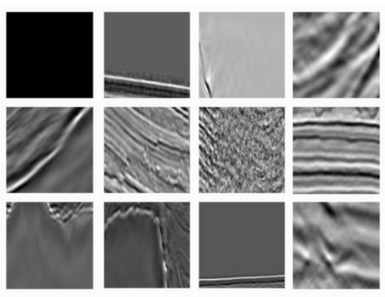
Haralick textural features for geological images segmentation

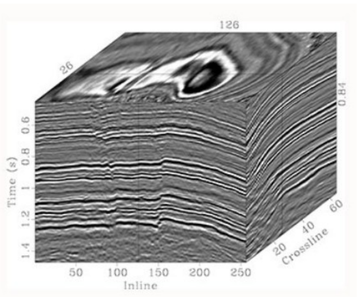
D. Kushnir[1]

Introduction

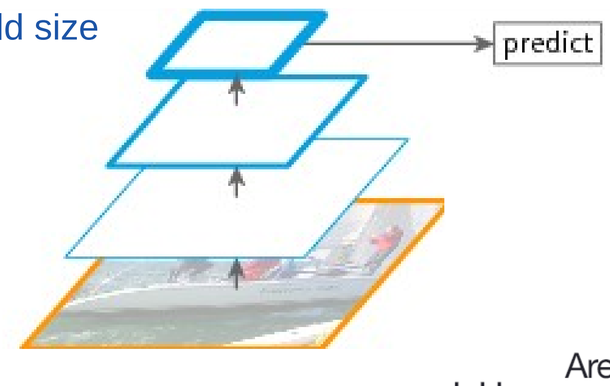
Several areas of Earth with large accumulations of oil and gas also have huge deposits of salt below the surface.

But unfortunately, knowing where large salt deposits are precisely is very difficult. Professional seismic imaging still requires expert human interpretation of salt bodies. This leads to very subjective, highly variable renderings. More alarmingly, it leads to potentially dangerous situations for oil and gas company drillers.





Different scale features used for larger perceptive field size



Metric used is Jaccard Similarity or so called “Intersection over union”, which is used for comparison of sets as well as image overlapping.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

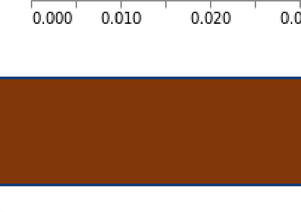
Kaggle-masters with fine-tuned CNN

#	△pub	Team Name	Score
1	—	b.e.s. & phalanx	0.896469
2	▲ 6	Tim & Sberbank AI Lab	0.895906
3	▲ 6	[ods.ai] topcoders	0.895456
4	▼ 2	SeuTao &CHAN &Venn &Keles	0.895439
5	▼ 1	Kent AI Lab | Ding Han | Renan	0.894755

… The One some student with

cold-runned XgBoost on GLCM

0.856856844...



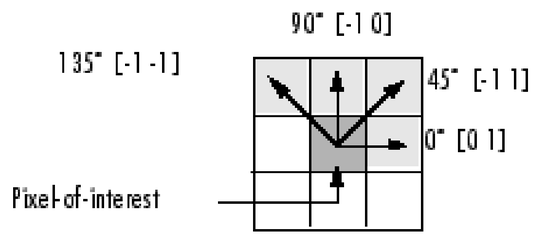
It showed up that the most important feature is ‘Energy’, it is a sqrt of Angular Second Momentum:

$$\text{Energy} = \sqrt{\text{ASM}}$$

$$\text{ASM} = \sum_i \sum_j p_d^2(i,j)$$

GLCM & Haralic

Image texture is a quantification of the spatial variation of grey tone values. Haralick et al. (1973) suggested the use of gray level co-occurrence matrices (GLCM). This method is based on the joint probability distributions of pairs of pixels. GLCM show how often each gray level occurs at a pixel located at a fixed geometric position relative to each other pixel, as a function of the gray level. An essential component is the definition of eight nearest-neighbor resolution cells that define different matrices for different angles (0°,45°,90°,135°) and distances between the horizontal neighboring pixels.

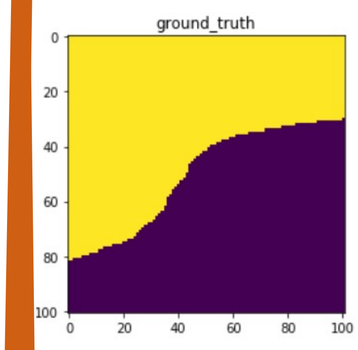


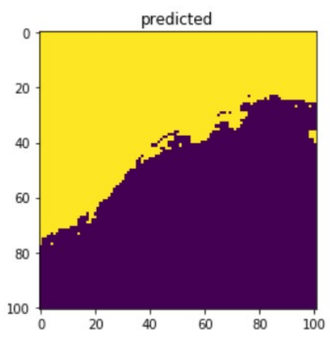
The methodology implies the combined use of three steps:

1st : Compute GLCM of the particular image and then extract Haralic features with different window size.

2nd: Use whole set of features on small subsample to estimate features’ F-values.

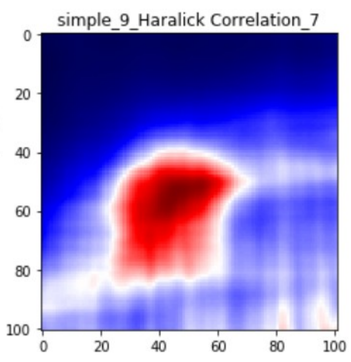
3rd: Use best features to train final model, as have less features can increase subsample size.

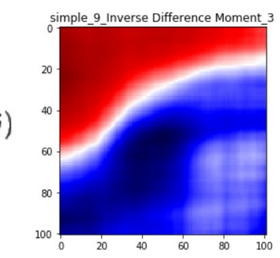




Feature examples:

$$\frac{\sum_i \sum_j (ij)p(i,j) - \mu_t^2}{\sigma_t^2}$$





How graycomatrix calculates several values in the GLCM of the 4-by-5 image I. Element (1,1) in the GLCM contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the values 1 and 1. Element (1,2) in the GLCM contains the value 2 because there are two instances in the image where two, horizontally adjacent pixels have the values 1 and 2. graycomatrix continues this processing to fill in all the values in the GLCM.

Conclusions

Category	CNN	GLCM
Data	4000 images in dataset considered as small	600 images is way too much.
Preprocessing	Augmenting images to regularize and increase train-set size. Normalizing is crucial for performance.	Creating of Haralic features from GLCM, pretty fast, but takes lot of space
Computing	GPU-farm would be nice	Feasible on PC CPU
Implementation	Trial and fail to find the best configuration, but search is guided by luck	Download libs, read guide, use python API
Batch learning	Batch-size is also hyper-parameter	Impossible, this part is disastrous
Features interpretability	Can visualize layers, but in case of sediments recognition it is useless	Can extract exact solving tree, range your features importance, etc...
Model preparation	The best-of-kind is proven to be U-net arcitecure build on SE-ResNext-50 pretrained on ImageNet	No retraining required, works out-of-box
Postprocessing	Result needs some smoothing of edges of masks to better fit with labeled by human	More random artifacts observed in this experiment, edges look better

- Statistical-based image preprocessing is still great for image
- Competitive data science and real-world tasks may not use the same notion of state-of-art

Future work:

- automated postprocessing
- exhaustive search of best parameters for generating features
- XGBoost fine-tuning

Applied Sciences Faculty, Ukrainian Catholic University, Lviv, Ukraine (¹kushnir_d@ucu.edu.ua)
GitHub: https://github.com/KushnirDmytro/salt_recognition_on_raw_images/