# TGS Salt Identification Challenge – Team Member 1: Marcus Ashiangmor

## 1. Feature Engineering Strategies

Data Augmentation:

Implemented four strength levels (light, medium, hard, harder) using Albumentations:

* Flips (horizontal, vertical)
* Brightness/contrast adjustments
* Gaussian noise, elastic and grid distortions
* Cutout, shear/scale transforms

1. Normalization:

* Added per-image z-score normalization (zero mean, unit variance)
* Employed in-place batch norm + activation (ABN) for stable feature distributions

1. Regularization:

Introduced auxiliary “salt-present” head with BCE loss (weight = 0.5)

1. Dimensionality Reduction:

* Built ASPP-style module to concatenate dilated convs and compress via 1×1 convs
* Used global average pooling for long-range context

1. Data Balancing:

* Filtered out uninformative examples (empty masks, artifacts, tiny regions) via custom drop\_some

## 2. Implementation Details

• Language & Libraries: Python 3.11, PyTorch, Albumentations, scikit-image

• Key Files:

* - lib/dataset.py – EDA, augmentations, normalization, balancing
* - lib/augmentations.py – Custom transforms (RandomContrastGray, AxisShear/Scale)
* - models/modules/abn\_inplace.py – In-place ABN + activation

## 3. Integration & Experiment Plan

1. Baseline: Jeimy’s flips-and-contrast pipeline → Val IoU 0.7694

2. Medium Augmentations: medium\_augmentations() + drop\_some(...) → measure IoU

3. Harder Augmentations: hard\_augmentations() → measure IoU

4. Auxiliary Loss: salt-presence head + combined loss → measure IoU

By swapping each component, identify which step gives the largest uplift over 0.7694 baseline.