Group - Gradient Descent

1. Executive Summary & Objective

The main task of the week was research, implementation, and testing of classical modalities of feature extraction, namely, SIFT, HOG, GLCM, LBP, and simple contour-based analysis. The aim was to develop a strong mathematical model of wheat heads that works well when they are trying to solve the green-on-green occlusion problem when using different densities and lighting conditions. Having examined the literature in the field of computer vision and experimented with those algorithms on a small set of our training images, we now have the architectural template of the downstream classification pipeline of the project.

2. Literature Review: Theoretical Foundations

To guide our implementation strategy, we conducted a literature review of fundamental object detection and texture analysis papers:

- **HOG (Dalal & Triggs, 2005):** We read Histograms of oriented Gradients of human Detection. The main finding made was that fine gradients, fine binning of orientation and overlap normalization of blocks are fundamental to the sound encoding of shapes, especially the ability to identify physical features without color-based encoding.
- **SIFT (Lowe, 2004):** We have looked at Distinctive Image Features of Scale Invariant Keypoints. Although they are strong at wide-baseline matching, literature indicates that sparse keypoint detectors will not always give a hit on highly articulated or biologically variable structures, such as wheat.
- **Texture (Ojala et al. & Haralick):** The theoretical basis of Local Binary Patterns (LBP) and Gray-Level Co-occurrence Matrices (GLCM) were reviewed to comprehend the mathematical process of isolating the granular micro-texture of wheat and the smooth texture of the foliage at the background.

3. Feature Extraction Modalities Evaluated

We tested the following methods on a random sample of training images, namely test their performance on difficult illumination (harsh shadows, blown-out sunlight) and dense vegetation :

1. **SIFT (Scale-Invariant Feature Transform) :** As it had been expected in terms of our literature review, SIFT was not the best in this particular task. Wheat heads do not have hard, repetitive geometrical corners. SIFT keypoints were not triggered uniformly in the convoluted canopy hence, could not be used in dense bounding-box localization.
2. **HOG (Histogram of Oriented Gradients) :** Our key shape descriptive was HOG. Using the mapping of gradient orientations with 8x8 pixel cells, HOG was able to encode the spiky contours

that appear in the high frequencies of the wheat awns. To provide dependence on wavy illumination, we used CLAHE (Contrast Limited Adaptive Histogram Equalization) as a preprocessing before computing gradients and normalizing the lighting variations.

3. **GLCM (Gray-Level Co-occurrence Matrix) :** We tested GLCM so as to have the statistical spatial relationships of pixels. GLCM overcomes this problem by offering a powerful macro-texture measure, contrast, correlation, and homogeneity, which aids the algorithm in learning the dense and overlapping structure of the wheat canopy over open soil.
4. **LBP (Local Binary Patterns) :** Micro-texture extraction was done using LBP. Through analysis of the immediate environment of each pixel, LBP will generate a uniform binary image that analytically separates the bumpy, granular structure of the wheat, separate to the more smooth structure of the leaves.
5. **Contour-Based Analysis (Color & Morphology) :** An HSV color-masking pipeline was designed, and it was able to identify the yellowish/tan color signature of the wheat. Morphological operations (Closing and Opening) were applied to remove the noise to obtain continuous object boundaries (contours). This contour analysis came in as a strong spatial attention mask in order to check on our texture and gradient features.

4. Empirical Validation and Pipeline Inferences

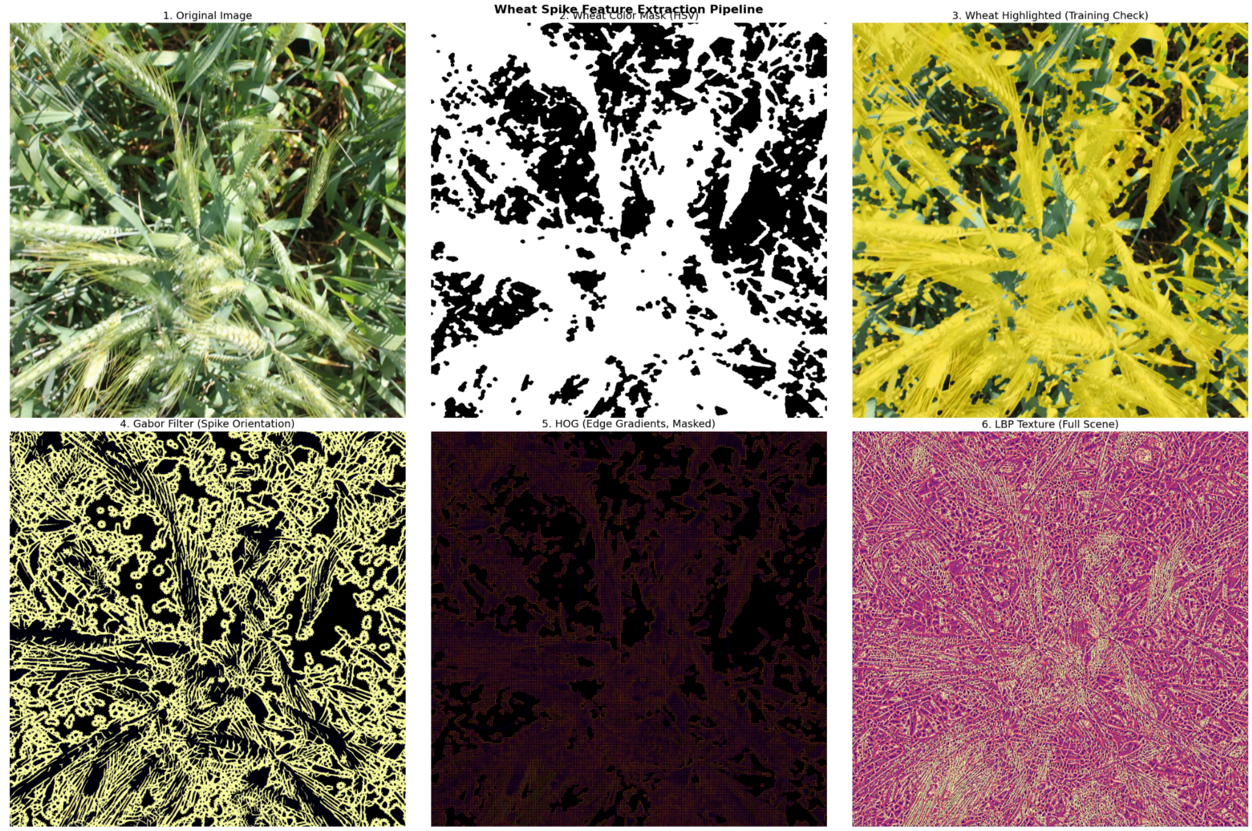
Based on our code execution on the training subset, we derive the following critical architectural inferences:

- **The Need of Feature Fusion:** There is no single feature description that can address the phenotyping issue. Color contours: under yellowing light Color contours do not work; HOG: under extreme level of wind blur GLCM/LBP: under scale variations. A combination of these (shape + textures + color contours) forms a very robust feature space.
- **Illumination Invariance:** It was empirically demonstrated that using CLAHE before HOG and LBP extraction, feature vectors in deep shadow regions are rescued, which meets the demands of the necessity to deal with changing lighting conditions.
- **Trap of HOG: Masking:** When used with a hard black background, it has been found by experiment that there are artificial gradient artifacts at the boundaries when HOG is applied after the mask. In the case of the next sliding-window version, HOG has to be calculated on the equalized unmasked grayscale picture to retain natural edge information.

Implementation and Output

We wrote the code to extract multiple feature layers from our dataset. We applied color masking, Gabor filters, HOG, and LBP to our wheat images.

The output we got after the implementation this week



Conclusion:

Our trials showed that any one of the mathematical characteristics on its own is a weak tactic when used in an agricultural setting. We have been able to assemble the structural geometry of HOG, granular micro-texture of LBP, and statistical macro-texture of GLCM, thereby creating a vastly robust, multi-dimensional representation of a wheat head.

Code:

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.feature import hog, local_binary_pattern
from skimage.filters import gabor
from google.colab import files

def process_wheat_for_training():
    print("Upload your wheat image:")
```

```python
uploaded = files.upload()

for filename in uploaded.keys():
    data = np.frombuffer(uploaded[filename], np.uint8)
    img = cv2.imdecode(data, cv2.IMREAD_COLOR)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # ── 1. COLOR-SPACE SEPARATION (Most important change) ──────────────────
    # Wheat spikes are yellowish/tan — HSV isolates them from green leaves
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

    # Mask for wheat spike color (yellowish-green to tan range)
    lower_wheat = np.array([15, 20, 80])
    upper_wheat = np.array([45, 180, 255])
    wheat_mask = cv2.inRange(hsv, lower_wheat, upper_wheat)

    # Also catch brighter/whiter spike tips
    lower_tip = np.array([0, 0, 180])
    upper_tip = np.array([180, 40, 255])
    tip_mask = cv2.inRange(hsv, lower_tip, upper_tip)

    combined_mask = cv2.bitwise_or(wheat_mask, tip_mask)

    # Morphological cleanup — connect broken spike regions
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
    combined_mask = cv2.morphologyEx(combined_mask, cv2.MORPH_CLOSE, kernel, iterations=2)
    combined_mask = cv2.morphologyEx(combined_mask, cv2.MORPH_OPEN, kernel, iterations=1)

    # ── 2. MASKED + ENHANCED GRAYSCALE ──────────────────────────────────────
    # Apply mask to original, then CLAHE only on wheat regions
    masked_img = cv2.bitwise_and(img, img, mask=combined_mask)
    gray = cv2.cvtColor(masked_img, cv2.COLOR_BGR2GRAY)

    clahe = cv2.createCLAHE(clipLimit=4.0, tileGridSize=(8, 8))
    gray_clahe = clahe.apply(gray)

    # ── 3. GABOR FILTER (Better than HOG for elongated spike detection) ─
    # Wheat spikes are elongated — Gabor captures oriented, linear structures
    gabor_responses = []
    for theta in [0, 30, 60, 90, 120, 150]:  # degrees
        theta_rad = np.deg2rad(theta)
        real, _ = gabor(gray_clahe, frequency=0.15, theta=theta_rad, sigma_x=3, sigma_y=6)
        gabor_responses.append(np.abs(real))
    gabor_combined = np.max(gabor_responses, axis=0)
    gabor_norm = cv2.normalize(gabor_combined, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)

    # ── 4. HIGH-RES HOG (kept but on masked image) ──────────────────────────
    features, hog_img = hog(gray_clahe,
                            orientations=12,
                            pixels_per_cell=(4, 4),   # 4x4 is more stable than 2x2
                            cells_per_block=(2, 2),
```

```
                        visualize=True)
hog_bright = np.power(hog_img / (hog_img.max() + 1e-8), 0.3)

# ── 5. LBP on original full gray (for background texture reference) ─
gray_full = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
clahe_full = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
gray_full_clahe = clahe_full.apply(gray_full)
lbp = local_binary_pattern(gray_full_clahe, P=16, R=2, method="uniform")

# ── 6. WHEAT OVERLAY (Training visualization) ──────────────────────
# Green overlay on detected wheat regions for annotation verification
overlay = img_rgb.copy()
overlay[combined_mask > 0] = (
    overlay[combined_mask > 0] * 0.4 + np.array([255, 220, 0]) * 0.6
).astype(np.uint8)

# ── DISPLAY
────────────────────────────────────────────────────────────────────────
fig, axes = plt.subplots(2, 3, figsize=(24, 16))

axes[0, 0].imshow(img_rgb)
axes[0, 0].set_title("1. Original Image", fontsize=14)
axes[0, 0].axis("off")

axes[0, 1].imshow(combined_mask, cmap='gray')
axes[0, 1].set_title("2. Wheat Color Mask (HSV)", fontsize=14)
axes[0, 1].axis("off")

axes[0, 2].imshow(overlay)
axes[0, 2].set_title("3. Wheat Highlighted (Training Check)", fontsize=14)
axes[0, 2].axis("off")

axes[1, 0].imshow(gabor_norm, cmap='inferno')
axes[1, 0].set_title("4. Gabor Filter (Spike Orientation)", fontsize=14)
axes[1, 0].axis("off")

axes[1, 1].imshow(hog_bright, cmap='inferno')
axes[1, 1].set_title("5. HOG (Edge Gradients, Masked)", fontsize=14)
axes[1, 1].axis("off")

axes[1, 2].imshow(lbp, cmap='magma')
axes[1, 2].set_title("6. LBP Texture (Full Scene)", fontsize=14)
axes[1, 2].axis("off")

plt.suptitle("Wheat Spike Feature Extraction Pipeline", fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()

# ── SAVE MASK FOR TRAINING ──────────────────────────────────────────────
# Save binary mask — can be used as segmentation ground truth
mask_filename = filename.replace('.jpg', '_wheat_mask.png').replace('.jpeg', '_wheat_mask.png')
```

```
        cv2.imwrite(mask_filename, combined_mask)
        files.download(mask_filename)
        print(f"Mask saved: {mask_filename} — use this as segmentation label for training!")

process_wheat_for_training()
```