# Joint Plant-Spraypoint Detector with ConvNeXt Modules and HistMatch Normalization

# Supplementary Material


## Authors:

**Ford, J.**

**Sadgrove, E.**

**Paul, D.**

**Figure S1: Lottes' model: (a)** A dense block containing a total of three convolutional layers. Each layer adds four filters to the feature map (L=3, G=4). **(b)** Macro structure showing the input image (green), the encoder (blue), the decoders (red) and the output feature maps (purple). Arrows display the flow of information through the model. Image resolutions presented are those given in the original paper.

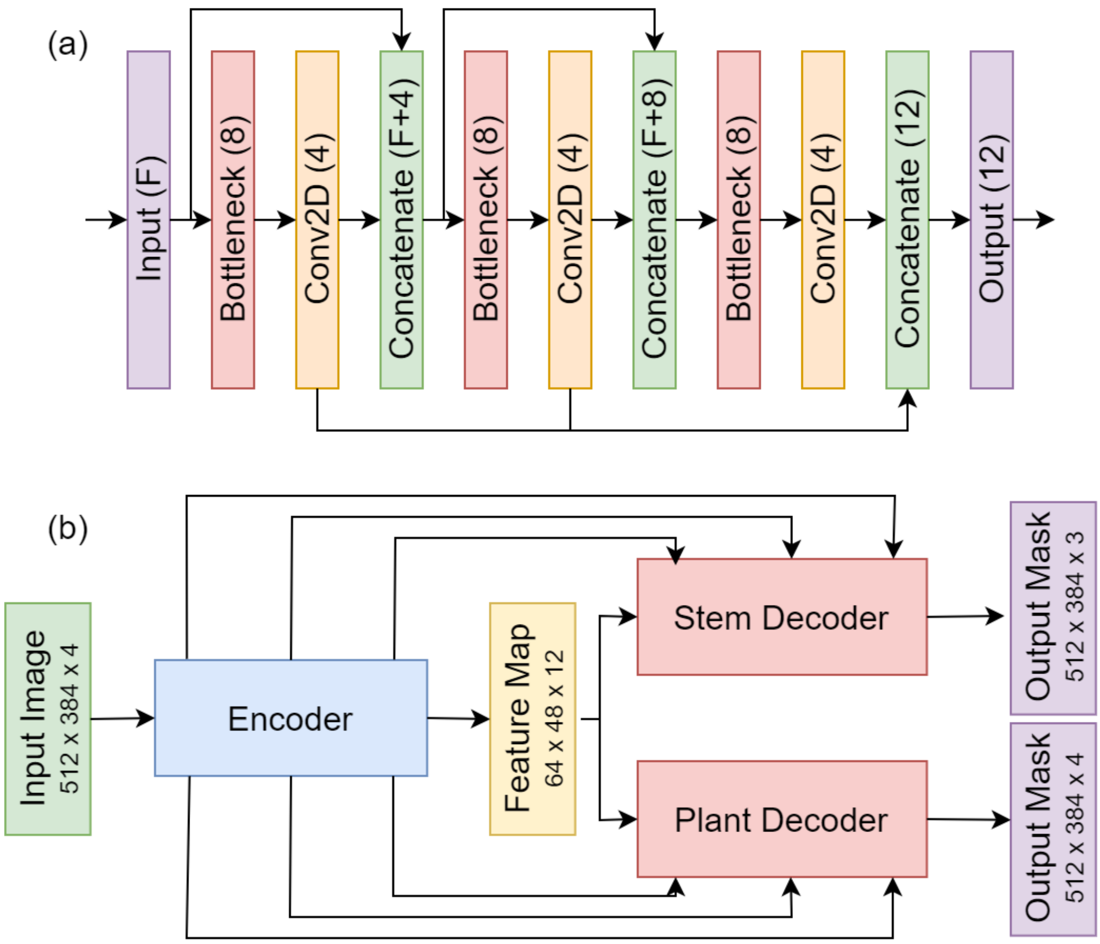

**Figure S2: Zhang's model: (a)** An IRM which employs two point-wise convolutions (orange), with a depth-wise convolution (red) in between. In the IRM, normalization occurs after each convolution, and activation occurs after the first point-wise convolution and the depth-wise convolution. **(b)** Macro structure showing the input image (green), the backbone (blue), the decoders (red) and the output feature maps (purple). Loss layers are shown in yellow, and include losses for deep supervision (small rectangles), as well as for the final outputs of the model (large rectangles). Arrows display the flow of information through the model. In particular, the cross-task feature fusion is given with a dashed line, and the bidirectional feature map exchanges are given with a dotted line. Image resolutions presented are those given in the original paper.

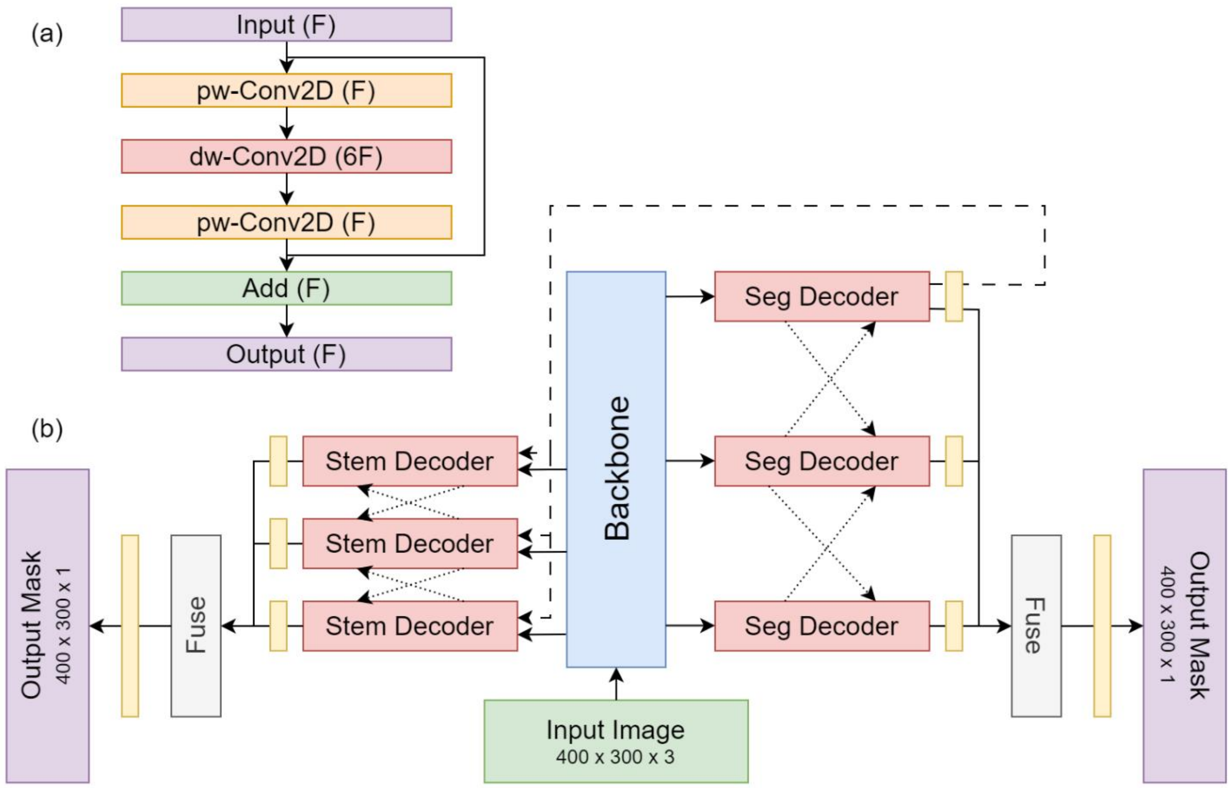

**Figure S3: Ours-4s model:** The 4s version of our model, which has one additional decoder for each tasks, which is situated at the 8x downsampling level for the spraypoint (or stem) task, and 4x downsampling level for the segmentation task. Components show are input image (green), patchify stem (orange), backbone (blue), decoders (red) and the output feature maps (purple). Loss layers are shown in yellow, and include losses for deep supervision (small rectangles), as well as for the final outputs of the model (large rectangles). Arrows display the flow of information through the model. In particular, the cross-task feature fusion is given with a dashed line, and the bidirectional feature map exchanges are given with a dotted line.

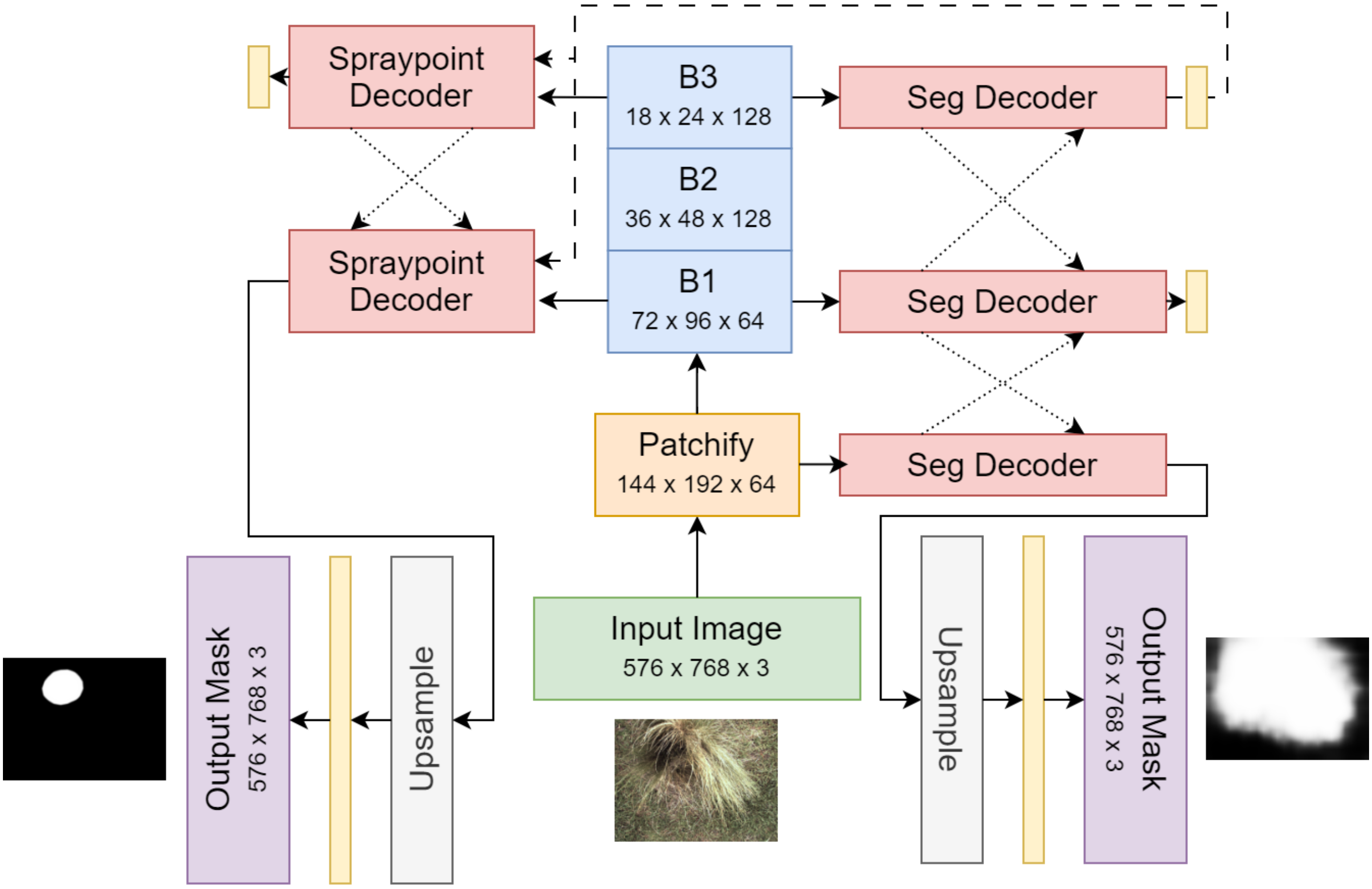

**Figure S4:** Normalized confusion matrices for the plant segmentation task for all models, for experiments 1, 2 and 3.

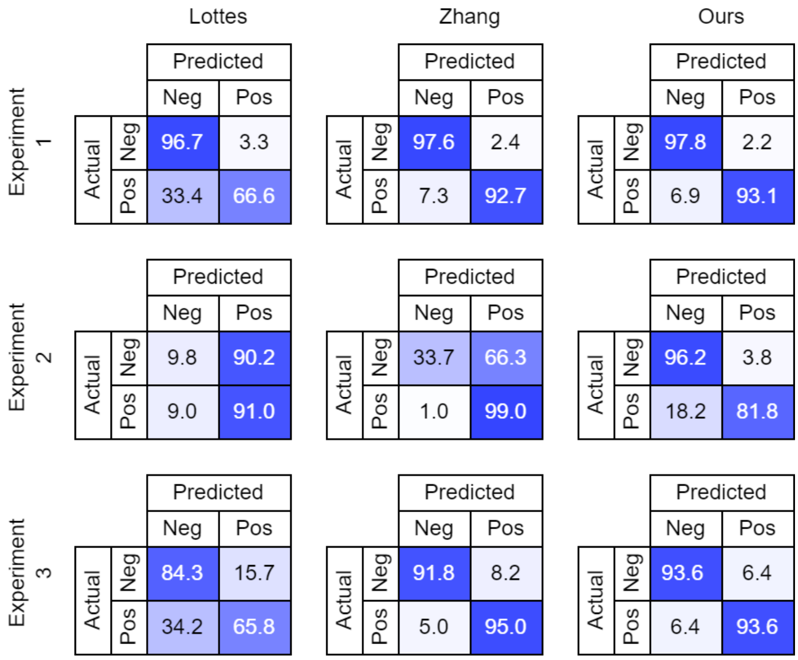

**Table S1: Experiment 1a metrics:**

| Task | Model | Normalization | | mIoU mean | mIoU std | Precision mean | Precision std | Recall mean | Recall std | F1 mean | F1 std |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Plant | Lottes | Image Norm | default | 0.874 | 0.014 | | | | | | |
| | | | std norm | 0.843 | 0.030 | | | | | | |
| | | | hist match | 0.871 | 0.010 | | | | | | |
| | | Batch Norm | default | 0.838 | 0.012 | | | | | | |
| | | | std norm | 0.624 | 0.041 | | | | | | |
| | | | hist match | 0.583 | 0.045 | | | | | | |
| | Zhang | Image Norm | default | 0.911 | 0.002 | | | | | | |
| | | | std norm | 0.913 | 0.001 | | | | | | |
| | | | hist match | 0.911 | 0.002 | | | | | | |
| | | Batch Norm | default | 0.909 | 0.001 | | | | | | |
| | | | std norm | 0.867 | 0.005 | | | | | | |
| | | | hist match | 0.862 | 0.008 | | | | | | |
| | Ours4 | Image Norm | default | 0.920 | 0.001 | | | | | | |
| | | | std norm | 0.918 | 0.002 | | | | | | |
| | | | hist match | 0.920 | 0.001 | | | | | | |
| | | Batch Norm | default | 0.913 | 0.003 | | | | | | |
| | | | std norm | 0.886 | 0.004 | | | | | | |
| | | | hist match | 0.879 | 0.002 | | | | | | |
| Spray | Lottes | Image Norm | default | 0.646 | 0.021 | 0.550 | 0.109 | 0.629 | 0.062 | 0.585 | 0.091 |
| | | | std norm | 0.656 | 0.007 | 0.682 | 0.060 | 0.631 | 0.028 | 0.653 | 0.028 |
| | | | hist match | 0.695 | 0.002 | 0.760 | 0.010 | 0.697 | 0.016 | 0.727 | 0.012 |
| | | Batch Norm | default | 0.603 | 0.013 | 0.452 | 0.047 | 0.437 | 0.062 | 0.441 | 0.038 |
| | | | std norm | 0.570 | 0.017 | 0.529 | 0.121 | 0.319 | 0.078 | 0.384 | 0.070 |
| | | | hist match | 0.543 | 0.017 | 0.457 | 0.101 | 0.181 | 0.046 | 0.252 | 0.051 |
| | Zhang | Image Norm | default | 0.800 | 0.005 | 0.851 | 0.015 | 0.900 | 0.022 | 0.875 | 0.014 |
| | | | std norm | 0.792 | 0.008 | 0.819 | 0.020 | 0.880 | 0.021 | 0.848 | 0.017 |
| | | | hist match | 0.792 | 0.007 | 0.825 | 0.026 | 0.890 | 0.012 | 0.856 | 0.016 |
| | | Batch Norm | default | 0.797 | 0.006 | 0.837 | 0.015 | 0.902 | 0.016 | 0.868 | 0.011 |
| | | | std norm | 0.787 | 0.003 | 0.833 | 0.039 | 0.846 | 0.025 | 0.838 | 0.009 |
| | | | hist match | 0.778 | 0.007 | 0.862 | 0.020 | 0.834 | 0.029 | 0.847 | 0.022 |
| | Ours4 | Image Norm | default | 0.830 | 0.003 | 0.879 | 0.020 | 0.905 | 0.016 | 0.892 | 0.015 |
| | | | std norm | 0.822 | 0.006 | 0.870 | 0.009 | 0.895 | 0.020 | 0.882 | 0.012 |
| | | | hist match | 0.822 | 0.002 | 0.878 | 0.012 | 0.903 | 0.018 | 0.891 | 0.009 |
| | | Batch Norm | default | 0.823 | 0.004 | 0.863 | 0.037 | 0.900 | 0.024 | 0.881 | 0.026 |
| | | | std norm | 0.812 | 0.005 | 0.871 | 0.017 | 0.854 | 0.020 | 0.862 | 0.017 |
| | | | hist match | 0.815 | 0.006 | 0.884 | 0.038 | 0.892 | 0.012 | 0.887 | 0.021 |

**Table S2: Experiment 2a metrics:**

| Task | Model | Normalization | | mIoU mean | mIoU std | Precision mean | Precision std | Recall mean | Recall std | F1 mean | F1 std |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Plant | Lottes | Image Norm | default | 0.376 | 0.077 | | | | | | |
| | | | std norm | 0.534 | 0.054 | | | | | | |
| | | | hist match | 0.703 | 0.052 | | | | | | |
| | | Batch Norm | default | 0.409 | 0.058 | | | | | | |
| | | | std norm | 0.402 | 0.037 | | | | | | |
| | | | hist match | 0.438 | 0.083 | | | | | | |
| | Zhang | Image Norm | default | 0.643 | 0.085 | | | | | | |
| | | | std norm | 0.810 | 0.016 | | | | | | |
| | | | hist match | 0.821 | 0.054 | | | | | | |
| | | Batch Norm | default | 0.654 | 0.150 | | | | | | |
| | | | std norm | 0.787 | 0.051 | | | | | | |
| | | | hist match | 0.813 | 0.025 | | | | | | |
| | Ours4 | Image Norm | default | 0.821 | 0.008 | | | | | | |
| | | | std norm | 0.826 | 0.039 | | | | | | |
| | | | hist match | 0.854 | 0.006 | | | | | | |
| | | Batch Norm | default | 0.820 | 0.016 | | | | | | |
| | | | std norm | 0.831 | 0.008 | | | | | | |
| | | | hist match | 0.841 | 0.005 | | | | | | |
| Spray | Lottes | Image Norm | default | 0.307 | 0.096 | 0.037 | 0.014 | 0.114 | 0.051 | 0.056 | 0.022 |
| | | | std norm | 0.577 | 0.010 | 0.325 | 0.059 | 0.454 | 0.028 | 0.376 | 0.041 |
| | | | hist match | 0.633 | 0.021 | 0.634 | 0.052 | 0.572 | 0.040 | 0.599 | 0.019 |
| | | Batch Norm | default | 0.429 | 0.085 | 0.064 | 0.041 | 0.154 | 0.034 | 0.086 | 0.039 |
| | | | std norm | 0.522 | 0.011 | 0.296 | 0.070 | 0.182 | 0.062 | 0.214 | 0.050 |
| | | | hist match | 0.528 | 0.016 | 0.274 | 0.078 | 0.192 | 0.069 | 0.214 | 0.063 |
| | Zhang | Image Norm | default | 0.690 | 0.026 | 0.717 | 0.101 | 0.595 | 0.051 | 0.646 | 0.058 |
| | | | std norm | 0.685 | 0.019 | 0.452 | 0.093 | 0.754 | 0.023 | 0.559 | 0.063 |
| | | | hist match | 0.734 | 0.010 | 0.735 | 0.043 | 0.732 | 0.039 | 0.733 | 0.037 |
| | | Batch Norm | default | 0.667 | 0.047 | 0.524 | 0.274 | 0.629 | 0.110 | 0.505 | 0.140 |
| | | | std norm | 0.702 | 0.010 | 0.652 | 0.077 | 0.675 | 0.014 | 0.660 | 0.034 |
| | | | hist match | 0.700 | 0.011 | 0.832 | 0.063 | 0.672 | 0.027 | 0.741 | 0.024 |
| | Ours4 | Image Norm | default | 0.747 | 0.005 | 0.882 | 0.045 | 0.637 | 0.017 | 0.740 | 0.025 |
| | | | std norm | 0.739 | 0.037 | 0.650 | 0.120 | 0.769 | 0.029 | 0.700 | 0.081 |
| | | | hist match | 0.776 | 0.008 | 0.843 | 0.023 | 0.772 | 0.034 | 0.806 | 0.028 |
| | | Batch Norm | default | 0.752 | 0.006 | 0.946 | 0.008 | 0.676 | 0.021 | 0.788 | 0.015 |
| | | | std norm | 0.783 | 0.004 | 0.871 | 0.019 | 0.758 | 0.020 | 0.810 | 0.014 |
| | | | hist match | 0.776 | 0.002 | 0.822 | 0.016 | 0.777 | 0.017 | 0.799 | 0.010 |

**HistMatch Normalization in Python (numpy):**

```
import numpy as np
from math import erf

# CDF for N(0,1)
def phi(x):
  return (1.0 + erf(x / 2.0**0.5))/2.0

# The number of points for which to calculate phi
N = 301

# [-3,3] will capture 99.7% of a standard normal distribution
yval = np.linspace(-3,3,N,endpoint=True)
ncdf = np.tile(np.array(list(map(phi,yval))).reshape(1,-1), (256,1))

# perform HistMatch normalization on an image.
def hist_match_norm(I):
    # Assume greyscale. Pixel values from 0 to 255
    # Normalize each channel separately for colour image

    # Cumulative histogram (normalized)
    h1 = np.sum(I.reshape(-1,1) <= range(256),0)
    h1 = h1/h1[-1]

    # Calculate midpoints
    h1 = (np.pad(h1,(1,0))[:-1] + h1)/2

    # Reshape for comparison
    h1 = h1.reshape(-1,1)

    # Find closest match between h1 and ncdf
    A = np.abs(h1-ncdf)
    f = yval[np.argmin(A,axis=1)]

    # apply function to I and return
    return f[I]

if __name__ == "__main__":
    # Display normalization graphs of test image (red channel)
    import PIL
    import matplotlib.pyplot as plt
    from matplotlib import gridspec

    I = np.array(PIL.Image.open("test.png"))
    I = I[:,:,0]

    J = (I - np.mean(I))/np.std(I)

    K = hist_match_norm(I)

    def snpdf(x):
        return (1 / np.sqrt(2*np.pi)) * np.exp(-0.5*x**2)

    x = np.linspace(-3,3,101,endpoint=True)
    p = snpdf(x)
```

```
fig = plt.figure(figsize=(6,8))
gs = gridspec.GridSpec(3,1,height_ratios=[4,1,1])

ax = [plt.subplot(g) for g in gs]

ax[0].imshow(I,cmap='gray')
ax[0].set_title('Image')
ax[0].set_axis_off()

ax[1].hist(J.flatten(),bins=50,density=True,alpha=0.6)
ax[1].plot(x,p,'k',linewidth=2)
ax[1].set_title('Naive')

ax[2].hist(K.flatten(),bins=50,density=True,alpha=0.6)
ax[2].plot(x,p,'k',linewidth=2)
ax[2].set_title('HistMatch')

plt.tight_layout()
plt.show()
```