Mass Estimation From Images Using Deep Neural Network and Sparse Ground Truth

Muhammad Hamdan
Department of Electrical and Computer Engineering
Iowa State University

Dec 19, 2019
Motivation
Sugarcane mass flow estimation methods

- Mass measurement through load cell [1, 2, 3, 4, 5, 6]
- Mass measurement through roller displacement [7]
- Volume measurement via optical sensor [8, 9]
- Volume measurement through roller displacement [7]

- Mass measurement through images from stereo camera

Inexpensive, simple, and relatively accurate
- Requires calibration and highly affected by changes in material density
- Depends on ambient light (night time and early morning)
Problem Complexity

• Factors
  – Angle of capture
  – Mass flow rate
  – Frame overlap
  – Variable elevator speed
  – Different run sizes
  – Different lighting conditions
  – Sparse ground truth
Deep Learning Basics

What to consider when deciding on using a DNN?

- CNN architecture
  - AlexNet, VGG, GoogleNet, ResNet, Your own?
- Activation function
  - Sigmoid, Tanh, ReLU, ELU
- Choice of hyper-parameters:
  - Learning rate
- Loss function
  - Classification: Softmax
  - Regression: MSE

$$\text{MSE} = L(y; \hat{y}) = \sum_{i=1}^{k} \frac{1}{n} (y_i - \hat{y}_i)^2$$
Loss Function

\[
L_i(x, y; w) = \frac{1}{n_i} \left\{ y_i - \sum_{j=1}^{n_i} (f(x_{ij}; w) \times v_{ij} \times t) \right\}^2
\]

\[
L_i(x, y; w) = \frac{1}{n_i} \left\{ y_i - \sum_{j=1}^{n_i} \hat{y}_{ij} \right\}^2
\]

Now that we handled frame overlap, we need to figure out how to obtain correct predictions per frame.
Gradient Update

• Our loss function
• Gradient update occurs at every end of a run
• We keep a running sum of gradients and predictions
• Compute the derivative of the loss function to apply loss

\[
\frac{\partial L_i}{\partial w} \leftarrow - \frac{2}{n_i} \left[ y_i - \sum_{j=1}^{n_i} \hat{y}_{ij} \right] \times \sum_{j=1}^{n_i} \frac{\partial \hat{y}_{ij}}{\partial w}
\]
**DNN Architecture Summary**

- **DNN Architecture**
  - **Input image size**: 96 x 144 (5th original size)
  - **Parameters**: K and Size of parameters: 0.17 MB
  - **Training time**: ~11 hours
  - **Testing average error**: 4.5%

**Res-9ER**

- **Input-Image RGB**: 7 x 7, 32
  - **Output size**: IN/2

- **CONV_1_1**: 1 x 1, 32
  - **Output size**: IN

- **CONV_2_1**: 1 x 1, 128
  - **Output size**: IN

- **CONV_2_2**: 3 x 3, 32
  - **Output size**: IN/2

- **CONV_2_3**: 1 x 1, 128
  - **Output size**: IN/2

- **CONV_1_2**: 3 x 3, 32
  - **Output size**: IN

- **CONV_1_3**: 1 x 1, 128
  - **Output size**: IN

- **Shortcut**: 1 x 1, 128
  - **Output size**: IN

**Questions?**
What is Going on Behind the Scenes?

• Proper visualization techniques can support the investigation of DNN functionality.
Histogram Distribution of Error and Outliers

Error Histogram Distribution

- DNN-Based Error: $\mu=0.021, \sigma=0.086$
- Volume-Based Error: $\mu=0.052, \sigma=0.112$
Questions

Questions?
References

Volumetric-Based Approach to Mass estimation

- Instant volume measurement is available
- Ground truth (true mass) is only available by run
- Ground truth (true mass) is only available by run

\[ Mass = f(max(V - \beta, 0); \theta) \times max(V - \beta, 0) \times v_{elev} \times t \]

Where "f" is a 2-layer neural network parameterized by "\theta" that outputs a prediction of density based on the volume (V), scaled by elevator speed \( V_{elev} \) and capture time (t), with tanh activation.
# Data Summary

## Laboratory data summary

<table>
<thead>
<tr>
<th>Location</th>
<th>Runs</th>
<th>Samples</th>
<th>Material</th>
<th>Representation</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISU</td>
<td>239</td>
<td>&gt;120K</td>
<td>Bamboo</td>
<td>Images and point cloud</td>
<td>Controlled</td>
</tr>
</tbody>
</table>

## Laboratory etup

![Laboratory setup image]

- **Camera & Lights**
- **Logging**
- **Bamboo Hopper**
- **Conveyor**
- **Scale**
Temporal Smoothness

- Images near in time should have more similarity in mass than images further away in time
- Hyper-parameter $\lambda$ (chosen empirically 0.05)
- This term is added to the loss function

$$L_i(x, y; w) = \frac{1}{n_i} \left\{ y_i - \sum_{j=1}^{n_i} (f(x_{ij}; w) \times v_{ij} \times t) \right\}^2 + \frac{\lambda}{n_i} \sum_{j=1}^{n_i} \left\{ f(x_{ij}; w) - f(x_i(j-1); w) \right\}^2$$