Classification loss: 

\[ \mathcal{L}_c = \min \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{2} 1(y^i = j) \log(p_{ij}) \]

Maximum Mean Discrepancy (MMD) [1] loss for domain adaptation:

\[ \mathcal{L}_M = \min \left\| \frac{1}{|X_a|} \sum_{x \in X_a} \Phi(x) - \frac{1}{|X_t|} \sum_{x \in X_t} \Phi(x) \right\|_2^2 \]

Overall Loss: 

\[ \mathcal{L} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{cs} + \lambda_2 \mathcal{L}_M \]

**MOTIVATION**

- Number of tampered images available to train a convolutional neural network is small.
- Inpainting and compositing, which are essentially forms of tampering similar to object removal and splicing, could be used to augment the data.
- Explore the possibility of performing domain adaptation between the augmented data as well as the curated data.

**APPROACH**

**TAMPERING DATA AUGMENTATION**

- Inpainting and compositing methods are employed.
- Three different augmentation schemes are used: Simple Inpainting, Semantic Inpainting and Feathering.
- Each of these schemes will help to augment at least one of copy-paste, object removal or splicing types of tampering.

**REFERENCES**