# Deep Learning Approaches for Determining Optimal Cervix Cancer Treatment

Chaitanya Asawa[1], Yushi Homma[2], Stuart Sy[1]

[1] Department of Computer Science, Stanford University, [2] Department of Mathematics, Stanford University

## Background & Motivation

- We aim to predict types of cervixes based on images
- Determining one's cervix type is crucial in determining what treatment is necessary at a pre-cancer stage
- This is currently a Kaggle competition; no past research.
- However, methods such as convolutional neural networks and transfer learning have found great success in image classification tasks

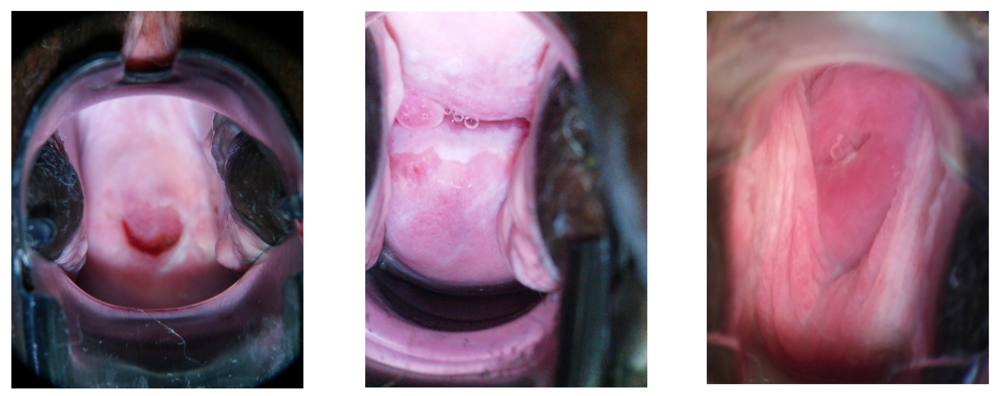
## Problem Statement

- Given an image of a cervix, we would like to output, for each class, the probability that the image belongs to this class
- We are investigating a combination of various CNN architectures, strategies such as Batch Normalization and Dropout, as well as transfer learning to obtain the best possible classification
- We will evaluate with the categorical cross entropy loss

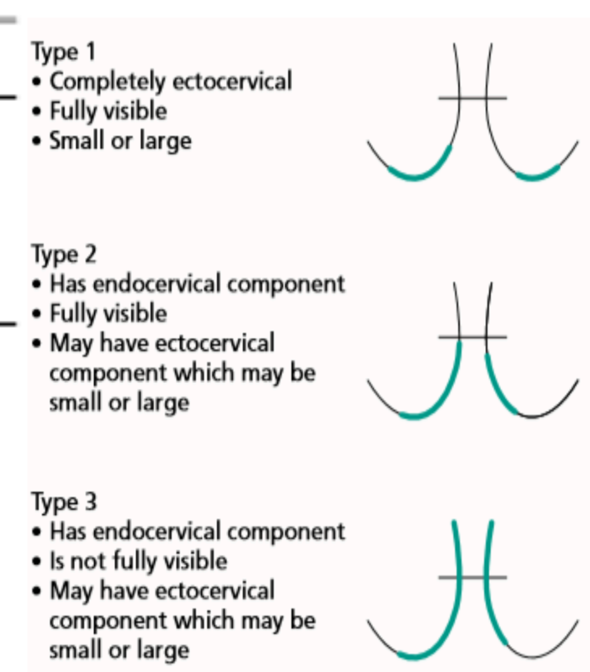
## Dataset

- Unbalanced dataset of cervix images, few hundred for 3 classes
- Various artifacts such as medical tools

| Type | Number of Images | Percentage |
|---|---|---|
| I | 250 | 17% |
| II | 781 | 53% |
| III | 450 | 30% |

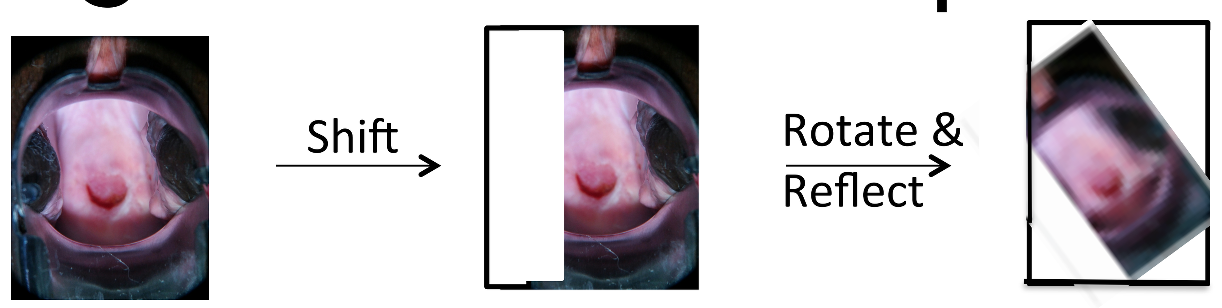


Type 1 Type 2 Type 3




Source: The Cervix, Singer et al, 2006

## Augmentation & Preprocessing




- We introduce more data and variations using shifts, rotations, and reflections of our original data
- We unit normalize our data using the training set's mean image and standard deviation

## Models & Methods

- Vanilla CNNs (just convolutional layers followed by affine layers)
- Retraining weights of successful ImageNet models
- Adding layers on top of successful ImageNet models
- Batch Normalization, Dropout, Cropping, Weighting

## Evaluation

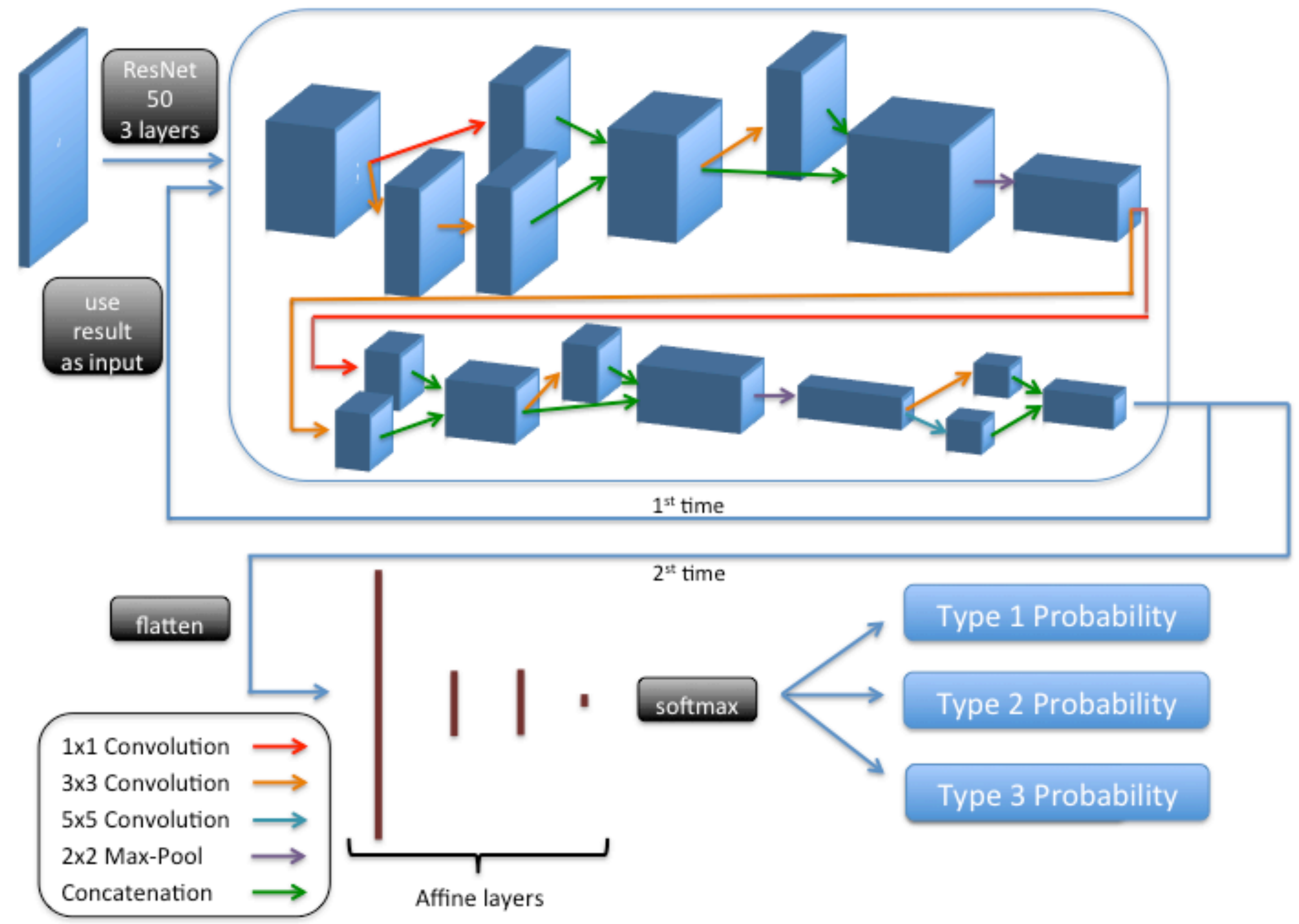
We will evaluate with the categorical cross entropy loss, which tries to maximize the predicted probability of the true class and minimize all others for an example.

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \ln \hat{y}_{ij}$$

*N*: The number of examples
*C*: The number of classes
$y_{ij}$: 1 if the *i*th example is of class *j*, 0 otherwise
$\hat{y}_{ij}$: Predicted probability *i*th example is of class *j*

## Inception-like Residual Network

The best performance we have been able to achieve has been using ResNet as an early feature extractor, followed by a series of Inception-like modules and residual connections.



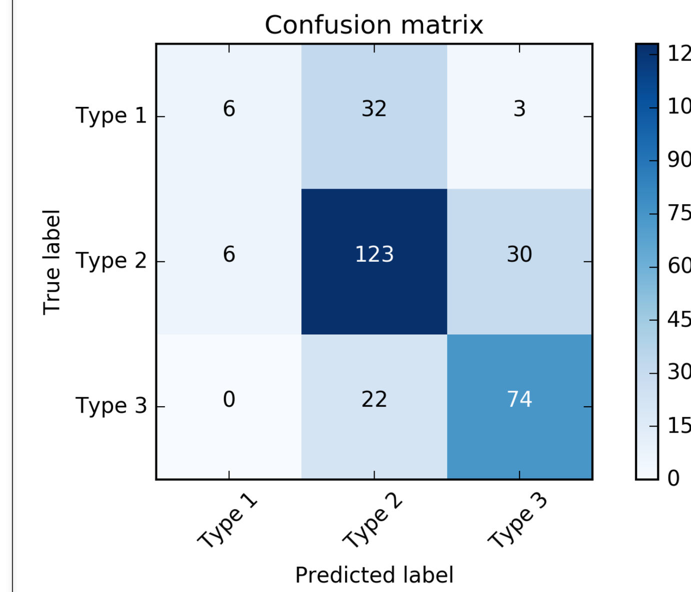

### Cropping Experiments

Transformation zone is usually closer to the center of the image, and so we aimed to focus there.

Slightly seems to help.

| % Centered | Cross Entropy | Accuracy |
|---|---|---|
| 100 | 0.8168 | 0.6318 |
| 85 | 0.8114 | 0.6453 |
| 70 | 0.7725 | 0.6824 |
| 55 | 0.8249 | 0.6115 |
| 40 | 0.8567 | 0.6014 |

### Confusion Matrix



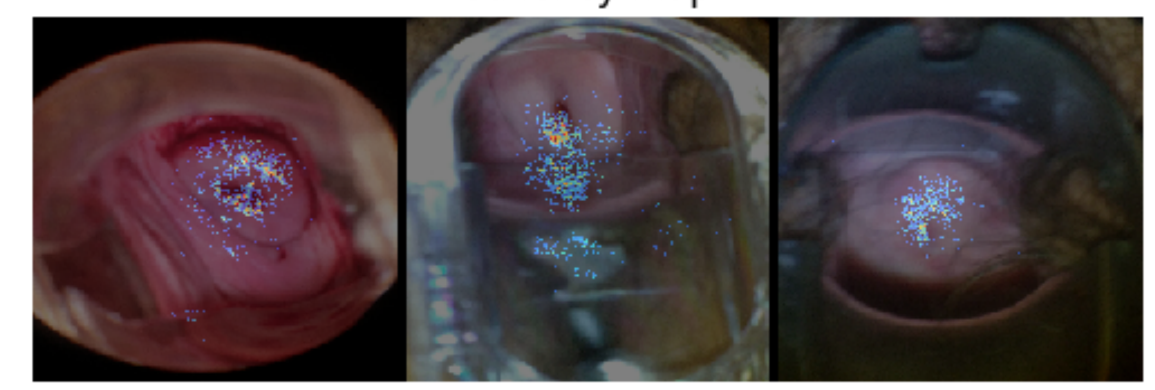

### Weighting Classes

| Weights (I, II, III) | Cross Entropy | Accuracy |
|---|---|---|
| 5.88,1.89, 3.33 | 0.9931 | 0.4831 |
| 4.88,1.89, 3.33 | 0.9696 | 0.5439 |
| 4.88,1.89, 2.33 | 0.9652 | 0.5608 |
| 2.5, 1.0, 1.5 | 0.8583 | 0.5777 |

Weighting does not seem to help our metric.

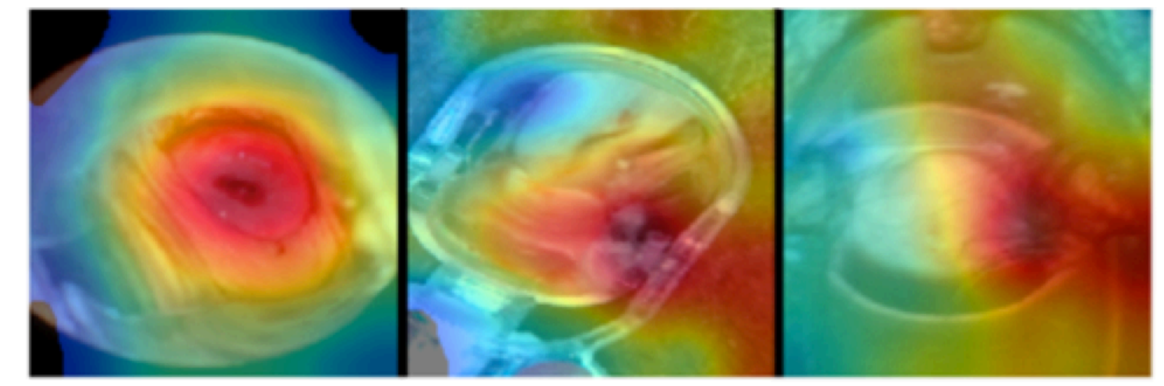
## Model Visualization

Saliency map



Different visualization methods show what regions of the image are most important when the model is making its prediction.

Class Activation Map



We successfully seem to make decisions based on the transformation zone.

## Comparison of All Models

| Model | Cross Entropy | Accuracy |
|---|---|---|
| Vanilla CNNs (no Inception-like modules) | 0.8618 | 0.5439 |
| Inception-v3 Transfer | 0.8417 | 0.6047 |
| ResNet Transfer | 0.7320 | 0.6858 |

We ultimately found that transfer learning was useful for feature extraction but required many new layers still to be adapted to our specific domain.

## Future Directions

Our results could perhaps be improved by first segmenting the transformation zone, the area of interest in our task, and then being able to classify based on the segmented region.