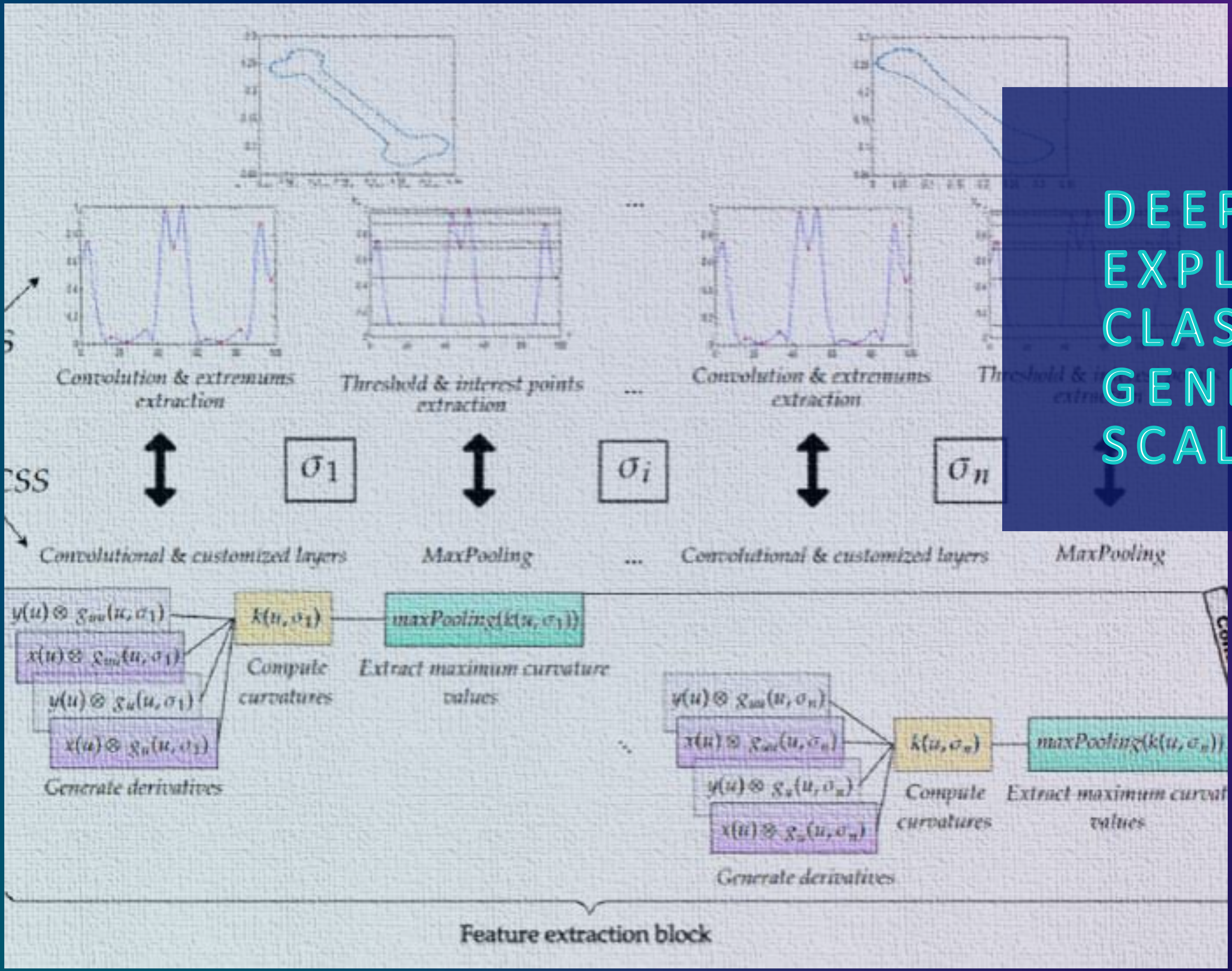


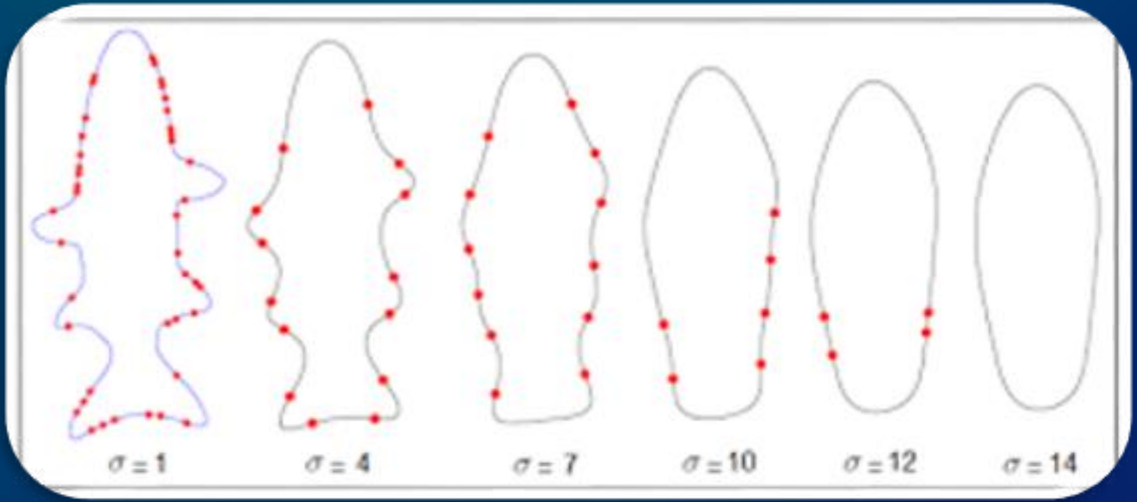
DEEPGCSS: A ROBUST AND EXPLAINABLE CONTOUR CLASSIFIER PROVIDING GENERALIZED CURVATURE SCALE SPACE FEATURES

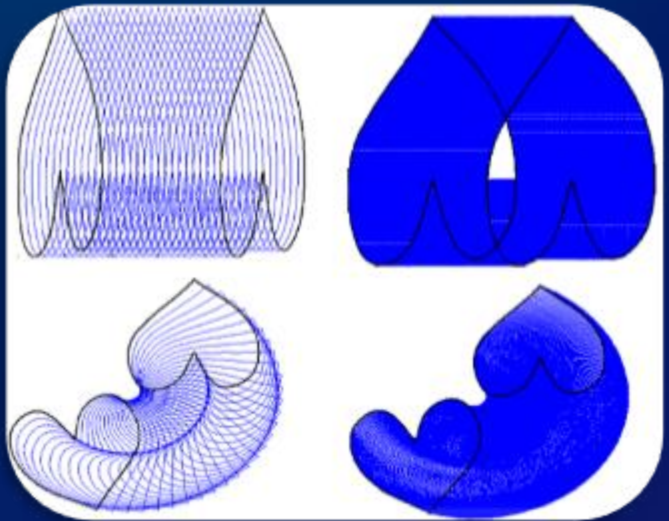
Group 3

Alecia DelCore,
Cole Rutledge,
Desmond Sheppard,
Darius Stevens,
Lance Swett,
Chantiese Tyler

KEY COMPONENTS OF DEEPGCSS

- Generalized Curvature Scale Space (GCSS) Descriptors
- Deep Neural Network
- Robustness
- Explainability





RELATED WORKS

ON ROBUSTNESS AND ACCURACY

“In our paper, we have proved that the accuracy metric is not enough to evaluate deep contour classifiers since the robustness that varies as a function of the attack is not necessarily correlated with the model accuracy.”

Paper

ROBUSTNESS

Measures the model's ability to maintain performance when faced with such challenges.

ACCURACY

Reflects how well a model performs under ideal conditions, but it does not indicate how the model handles challenges such as attacks, distortions, rotations, noise, or other disruptions.

ON TYPES OF SHAPE DESCRIPTORS

- Region based
- Contour based
- CSS Specific

Math Intro: CSS in 3 Steps

1. Arc Length Parameterization
2. Smoothing
3. Curvature

$$\Gamma : [0,1] \mapsto \mathbb{R}^2 \qquad t \mapsto [x(t), y(t)]^T$$

$$\Gamma^*(s) = [x(\phi^{-1}(s)), y(\phi^{-1}(s))]^T$$

$$\phi(u) = s(u) - s(0) = \int_0^u \Gamma'(u)\mathrm{d}u$$

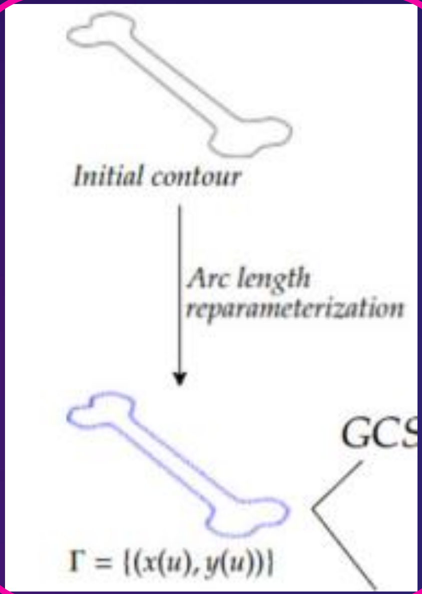
$$g(s,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-s^2}{2\sigma^2}}$$

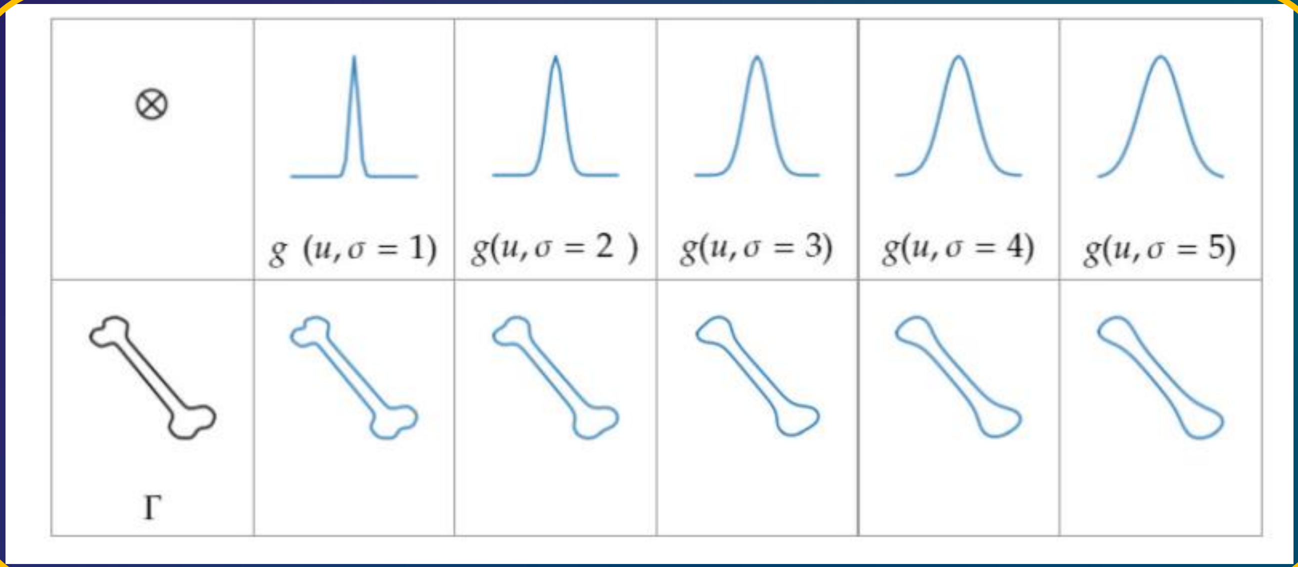
$$x(s,\sigma) = x(s) \otimes g(s,\sigma)$$

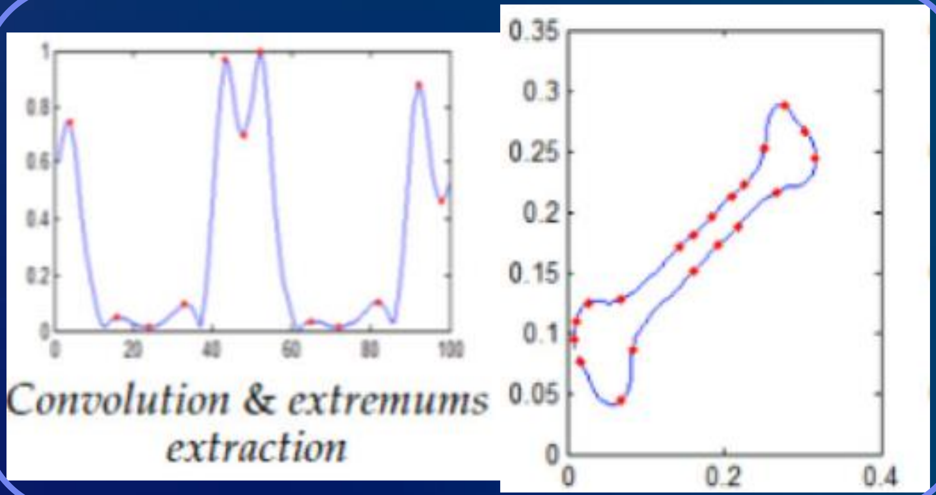
$$y(s,\sigma) = y(s) \otimes g(s,\sigma)$$

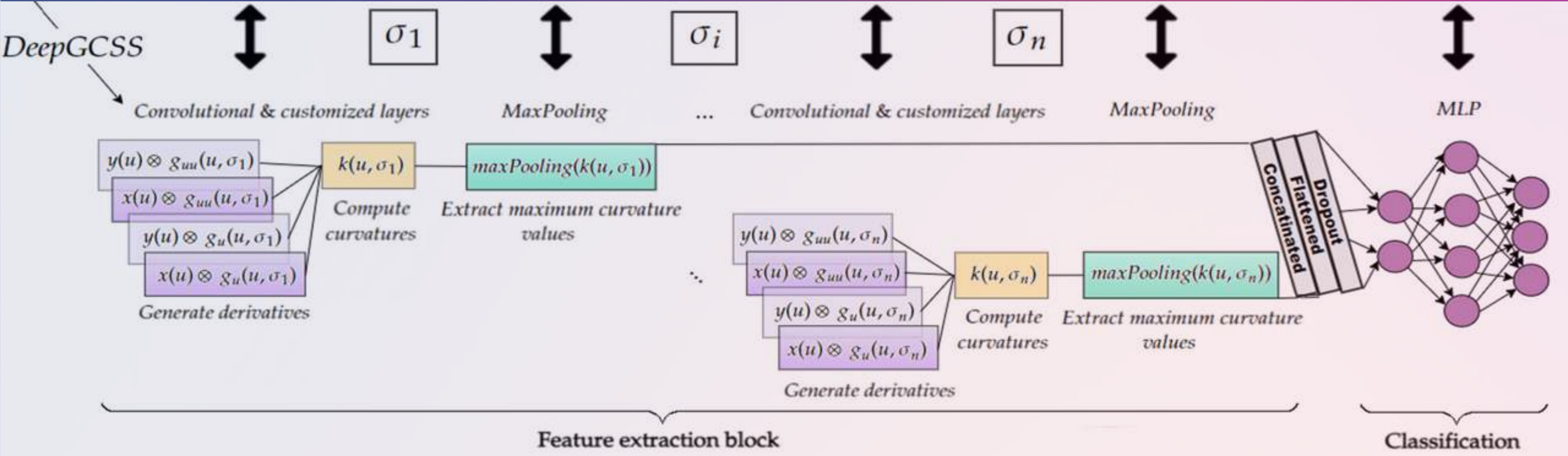
$$\kappa(s,\sigma) = \frac{\dot{x}(s,\sigma)\ddot{y}(s,\sigma) - \dot{y}(s,\sigma)\ddot{x}(s,\sigma)}{(\dot{x}^2(s,\sigma) + \dot{y}^2(s,\sigma))^{3/2}}$$

$$\kappa(s,\sigma) = \dot{x}(s,\sigma)\ddot{y}(s,\sigma) - \dot{y}(s,\sigma)\ddot{x}(s,\sigma)$$


Initial contour
Arc length reparameterization
GCS
$\Gamma = \{(x(u), y(u))\}$


$\otimes$
$g\ (u, \sigma = 1)$
$g(u, \sigma = 2\ )$
$g(u, \sigma = 3)$
$g(u, \sigma = 4)$
$g(u, \sigma = 5)$
Γ


Convolution & extremums extraction
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.2
0.4



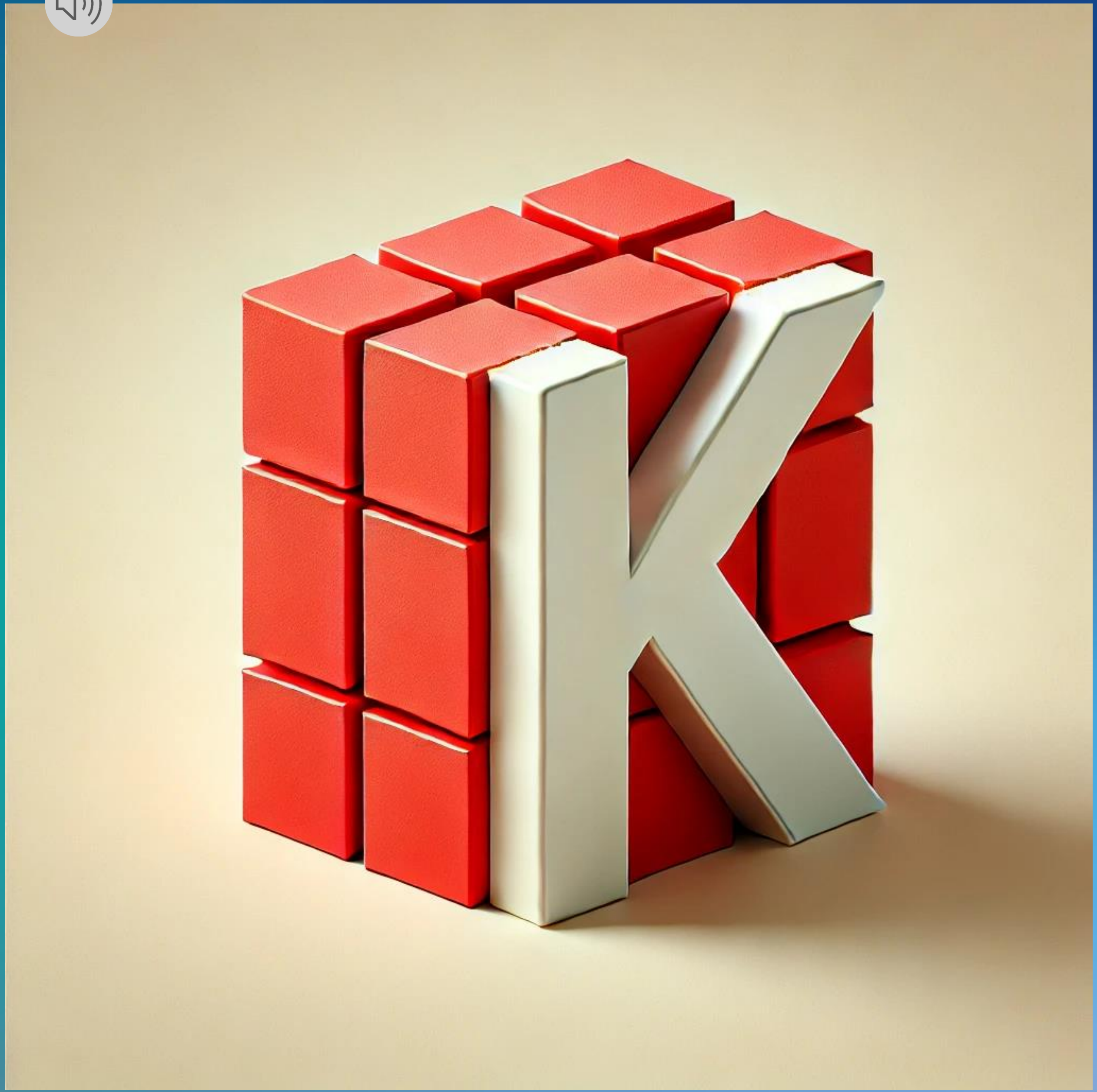
DeepGCSS

Contour Detection:
•The process begins with detecting the contours of objects in an image, then parameterizing the contour.

Feature Extraction:
•The GCSS descriptors are computed for the detected contours via curvature formula and MaxPooling.

Feature Learning:
•The extracted features are fed into the deep neural network.

Classification:
•Once trained, the network can classify new contours by analyzing their GCSS descriptors.



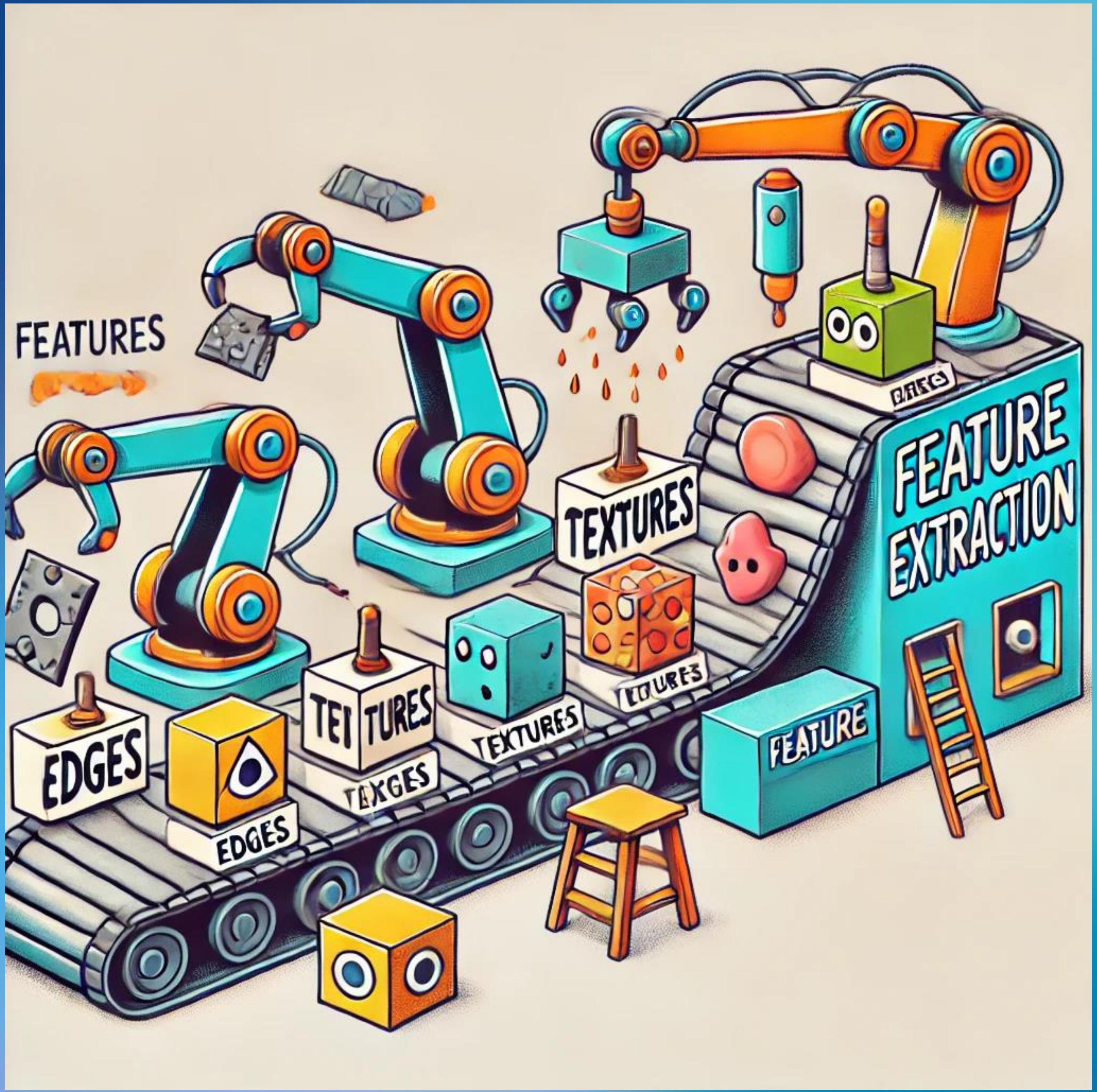
BUILDING BLOCKS WITH KERAS

- Input Layer
- Conv2D
- Pooling Layer
- Batch Normalization
- Dense Layers

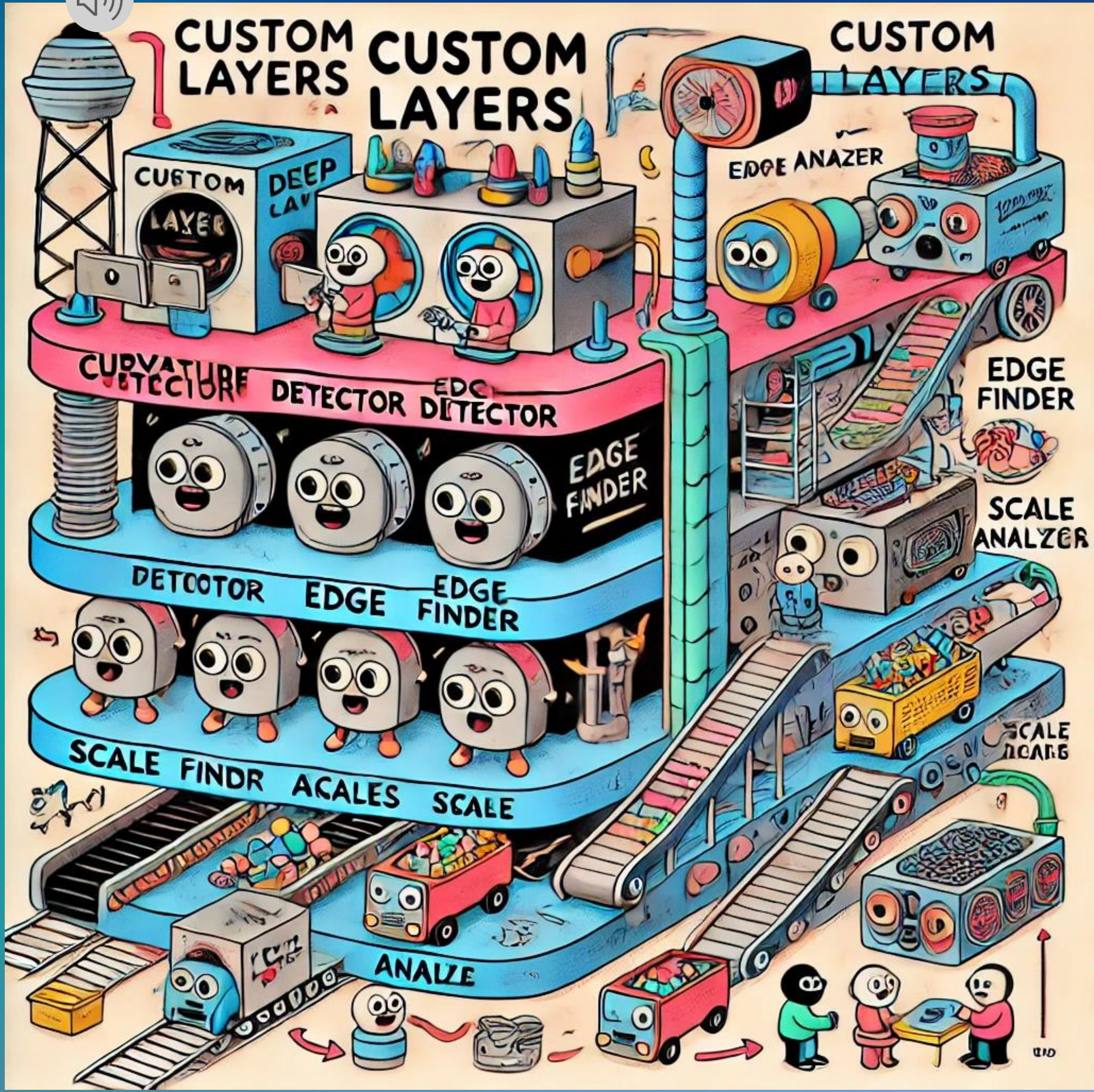
Speaker: Lance Swett

THE GCSS PART

- Purpose
- GCSS Mimic
- Convolution Layers
- Multiple Scales



Speaker: Lance Swett



CHALLENGES AND CUSTOM LAYERS

- Hard to replicate GCSS features
- Custom Layers Are a must
- Balance

- Keras defining custom processing

Speaker: Desmond Sheppard

UNIQUE METHODS

Traditional GCSS:
- Handcrafted Features
- Interpretable

Deep Learning:
- Automated feature extraction
- powerful

DeepGCSS:
- Best of both worlds

Speaker: Chantiese Tyler

RESULTS

MNIST

- 70K handwritten digits
- 60K used for training
- 10K used for validation

MPEG – 7

- Shapes
- 70 types of objects with 20 different shapes
- Total of 1,400 shapes

Fig. 2 MNIST Dataset: On the top, samples from the MNIST image dataset; On the bottom, corresponding samples for MNIST contour dataset

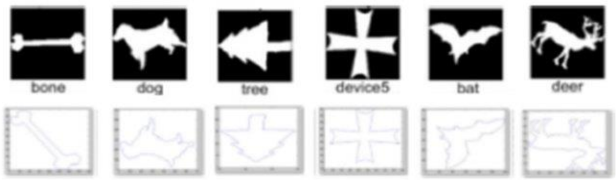
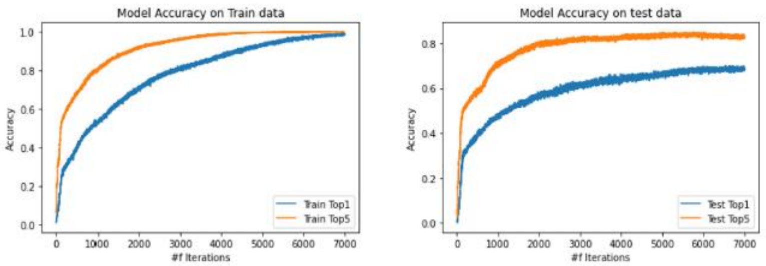


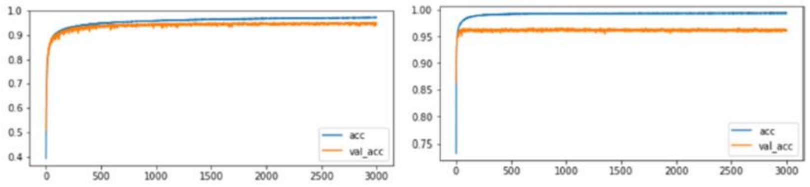
Fig. 3 MPEG-7 Dataset: On the top, some samples from MPEG-7 image dataset; On the bottom, samples from MPEG-7 contour dataset



(a) At epoch 7000, TOP5 train accuracy is equal to 100%; TOP1 train accuracy is equal to 98.98%

(b) At epoch 7000, TOP5 validation accuracy is equal to 82.14%; TOP1 validation accuracy is equal to 68.57%;

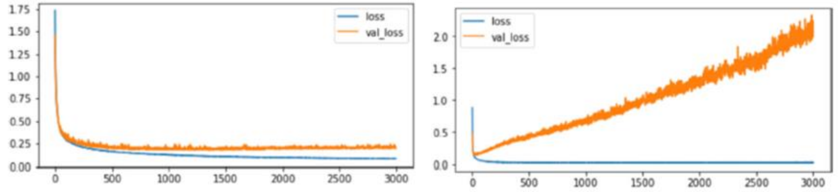
Fig. 7 DeepGCSS TOP1 and TOP5 accuracy on MPEG-7 contours dataset



(a) Train and validation accuracy of DeepGCSS: At epoch 3000, train accuracy=97.20%; validation accuracy=95.15%;

(b) Train and validation accuracy of DeepGCSS in case of random weights: At epoch 3000, train accuracy=99.32%; validation accuracy=96%

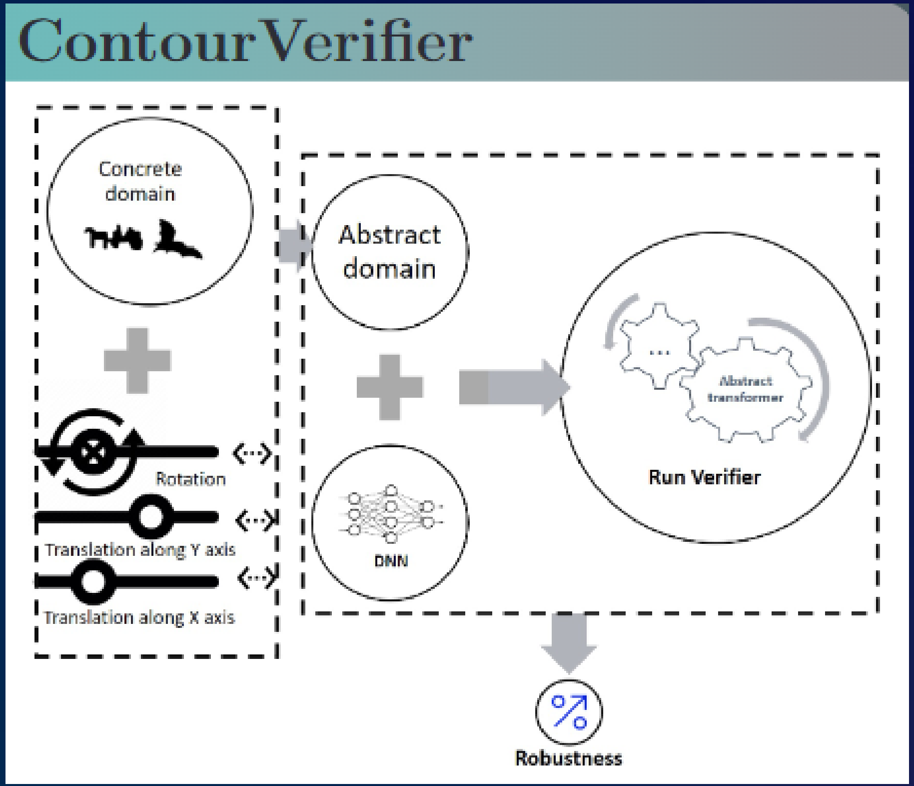
Fig. 5 Train and validation accuracy on MNIST digits contour dataset



(a) Train and validation loss of DeepGCSS: At epoch 3000, train loss=8.37%; validation loss=19.58%

(b) Train and validation loss of DeepGCSS in case of random weights: At epoch 3000, train loss=2.42%; validation loss=198%

Fig. 6 Train and validation loss on MNIST digits contour dataset



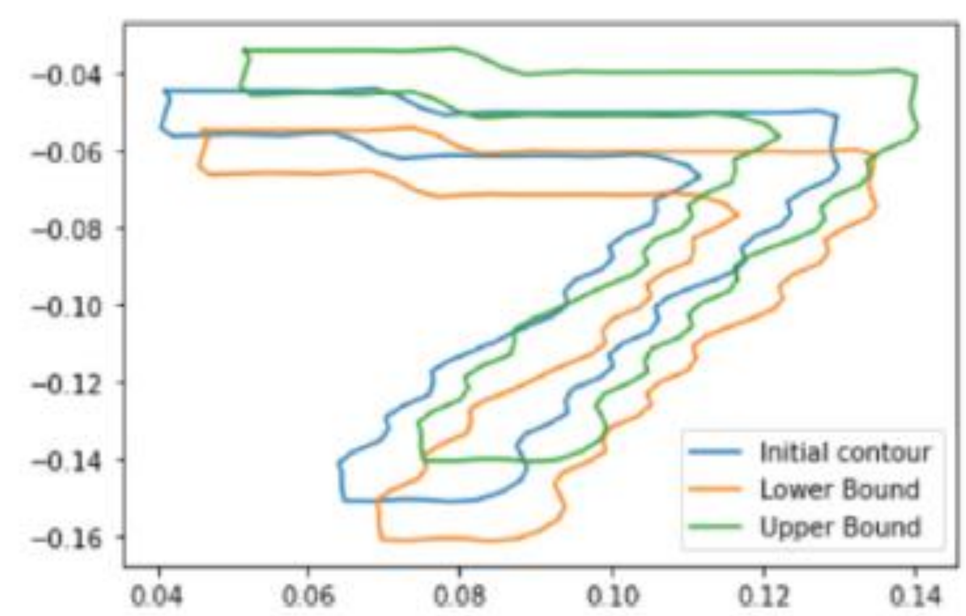


Fig. 8 Lower & Upper Bounds for translated contour (for dim1 = 0.005 and dim2 = 0.01) along x and y axis

ROBUSTNESS

ROBUSTNESS

Comparative Study Results:

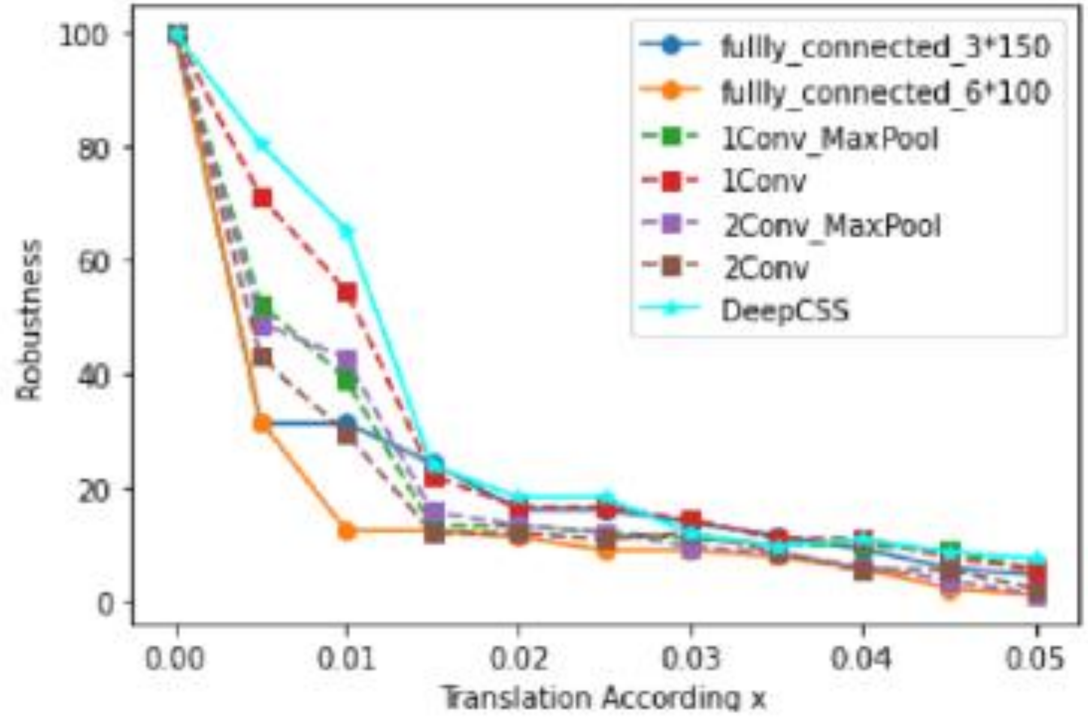
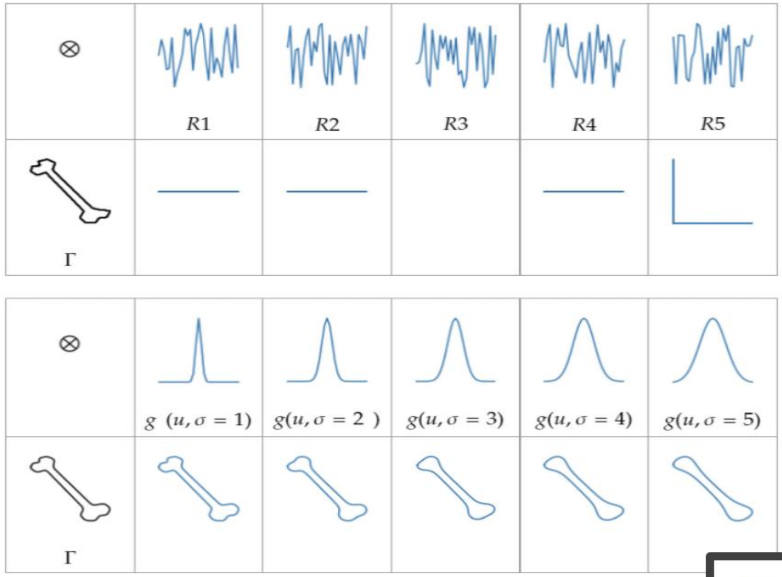


Fig. 9 Robustness variation according to the translation computed on 100 contours from MNIST contours dataset with different models including DeepGCSS

Speaker: Darius Stevens

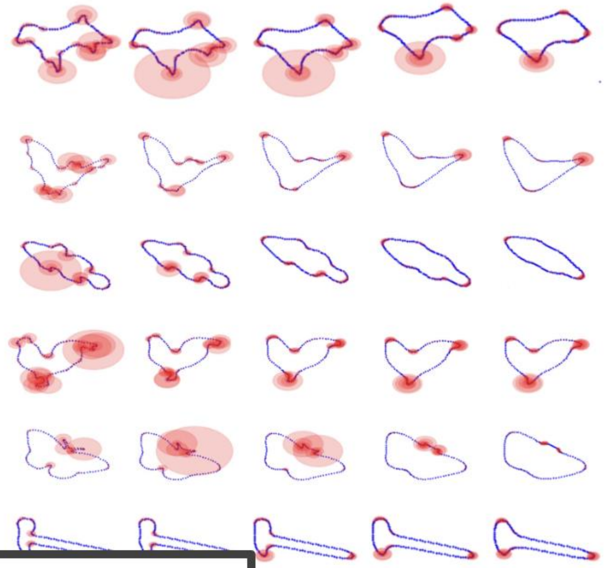


RANDOM Kernels

PREDEFINED Kernels

* Gaussian Kernel (formula 4) :

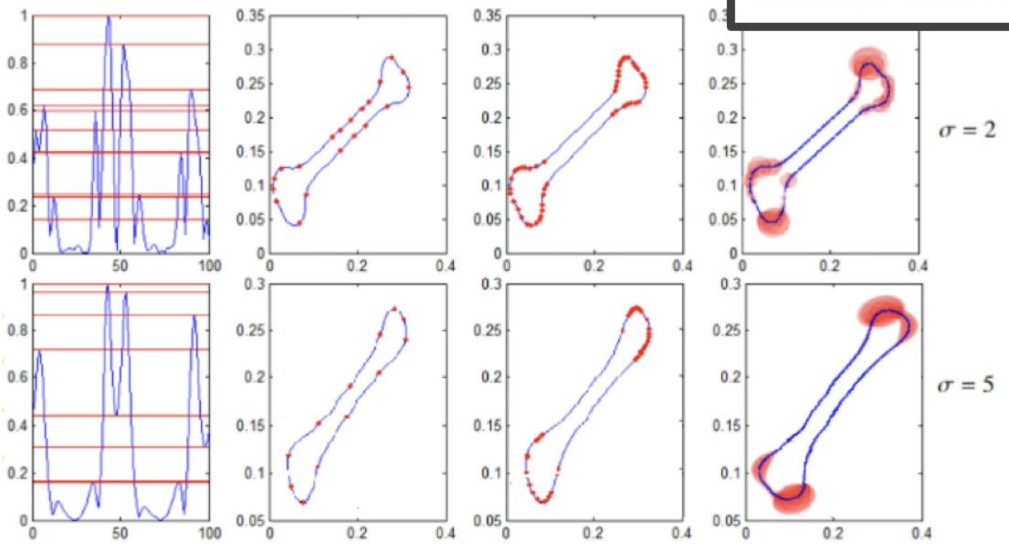
$$g(s,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-s^2}{2\sigma^2}}$$



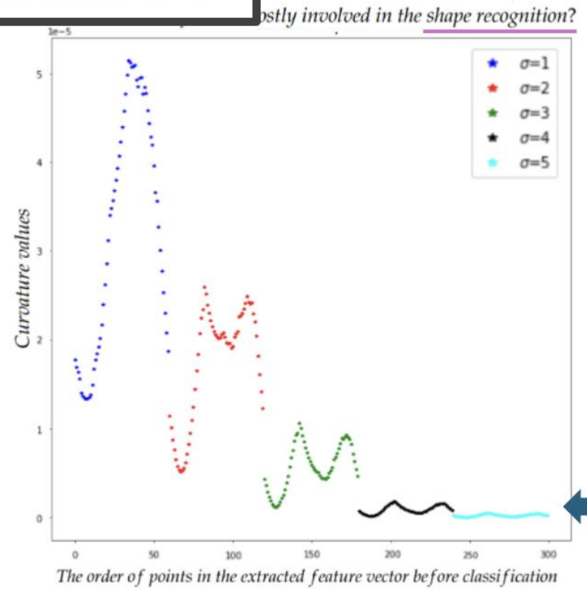
DeepGCSS Feature Extraction

highlights areas with strong/ sharp curves (explainable)

EXPLAINABILITY



CSS **GCSS** **DeepGCSS**



Key Points in CSS Representation:

Thresholding → max pooling

Scale optimization → autonomously optimizes sigmas

Dropout Layer

Speaker: Desmond Sheppard

APPLICATIONS OF DEEPGCSS

BETTER FOR:

Image / Video Analysis

Medical Image Analysis

Remote Sensing

Robotics / Computer Vision

(Loose Latency Requirement)

WORSE FOR:

Fine-Detail Analysis

Analysis Under Inconstant Lighting

Obscured Object Recognition

Real-Time Image Processing

(Strict Latency Requirement)

HYPOTHETICAL IMPROVEMENTS AND EXTENSIONS

DESCRIPTOR CHOICE

TYPES AND SCENARIOS

- Choice of descriptor can drastically change the accuracy, robustness, and computational cost of running the model
- Should be determined by computational cost and by application

EFFICIENCY

SYMMETRY ANALYSIS

- Implement a symmetry sorting algorithm
- Different paths for descriptors
- Model prefers Gaussian for less symmetric contours and Fourier for contours with higher radial symmetry

ROBUSTNESS

TOPOLOGY

- Utilize a topological descriptor like skeletonization or persistent homology
- Combine with gaussian smoothing and gaussian curvature descriptor
- Result: a model that very effectively can map a contour's global and local features

PARAMETERIZATION

AFFINE PARAMETERIZATION

- Different parameterization type
- Introduces curvature as a metric to determine how a contour is parameterized
- Adds a density component to parameterization
- Makes simple contours more unique

Speaker: Darius Stevens

WHAT WE LEARNED FROM OUR DL PROJECT

"My favorite part of the paper/class was getting to learn about math that I wouldn't necessarily choose to (Gaussian smoothing, etc) and learning about the affine parameterization vs euclidean."
- Cole Rutledge

"I was shocked that applying predefined weights will result in a more robust system. It showed me the limitations of AI. I would like to explore how this idea relates to other areas of deep learning."
- Darius Stevens

"I learned robustness does not necessarily correlate with accuracy."
- Alecia DelCore

"I gained a deeper understanding of how mathematical concepts like Gaussian kernels can be applied to solve real-world problems in computer vision."
- Des Sheppard

"I learned about accuracy of algorithms doesn't always correlate to real-world results. My favorite part would be understanding the implementation of tensor flow and keras." - Lance Swett

"I had a hard time understanding smoothing with Gaussian kernels. And why the smoothing was reducing how clear the images are."
- Chantiese Tyler

QUESTIONS?