Enhancing Accelerator Diagnostics

Real-Time ML-Driven Image Analysis

Prachiti Chandratreya (IIT Jodhpur)
DESY Summer Student Programme (MXL Group)

01.09.2025

Supervised by
Dr. Frank Mayet (DESY)
Dr. Sergey Tomin (DESY) and
Dr. Jitendra Kumar (IIT Jodhpur)

HELMHOLTZ

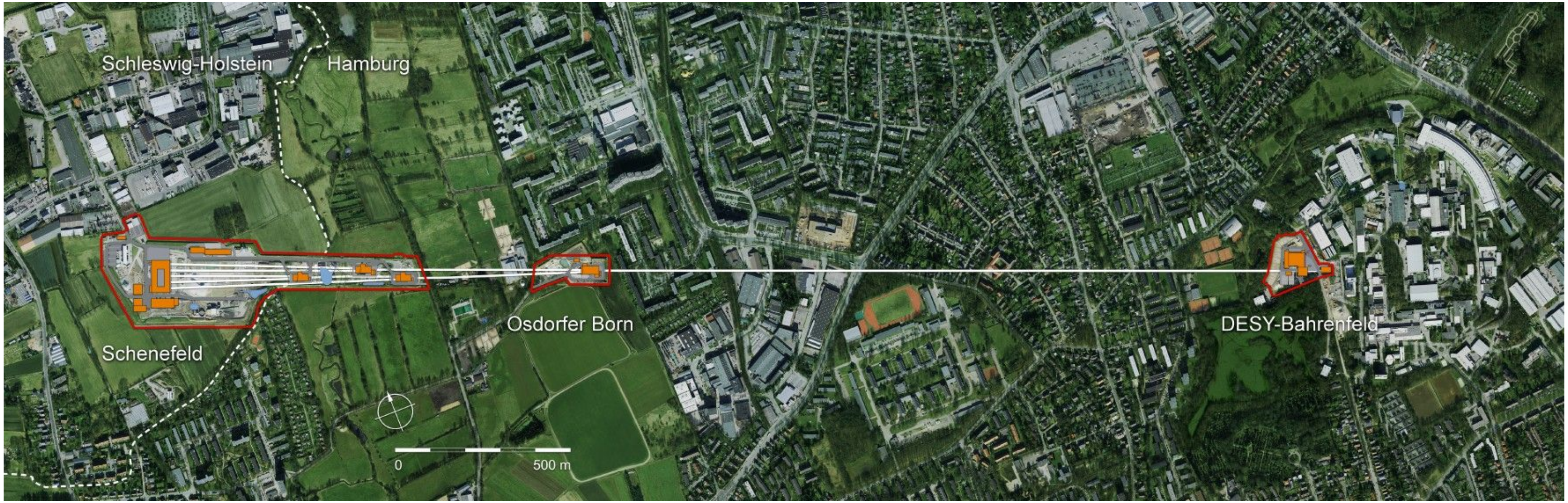
Outline

- ❑ XFEL Accelerator
- ❑ Motivation
- ❑ Accelerator Diagnostics review
- ❑ New methods for Diagnostics
  - ❑ Conventional
  - ❑ ML Based
- ❑ Status, summary and future plans

European XFEL Accelerator

https://www.xfel.eu/organization/mission/index_eng.html

- Located at DESY, operational since 2017 and produces **world's brightest X-ray laser**
  – ultrashort (<100 fs), highly coherent X-ray pulses, up to 27K per second
- **Enables breakthroughs in science** capturing ultrafast processes, imaging nanoscale structures, and probing matter under extreme conditions.



XFEL: Production of X-ray laser

Stage 1. Electrons (“seeds” of X-rays) are generated from **Electron Source.**

Stage 2**:** Electrons **are accelerated up to 17.5 GeV** using a superconducting LINAC accelerator.

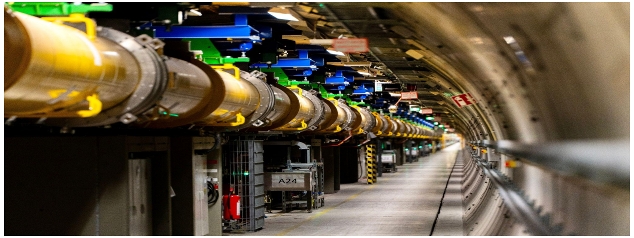
Stage 3: Magnetic electron bunch compressors → squeeze **beam to few micrometers**

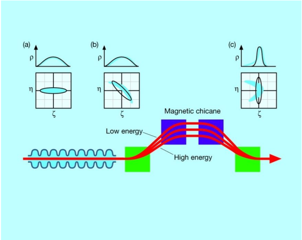
Stage 4: Electron beam enters undulators (arrays of alternating magnets)

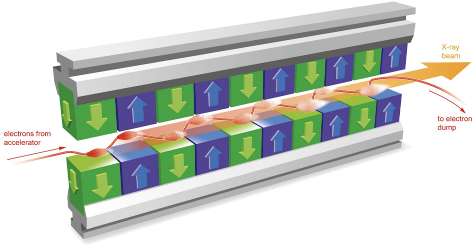
→ **X-ray pulses** of ultrashort duration (<100 femtoseconds), extremely bright, and coherent

→ **Scientific Studies**

Electrons Gun







Stage 4

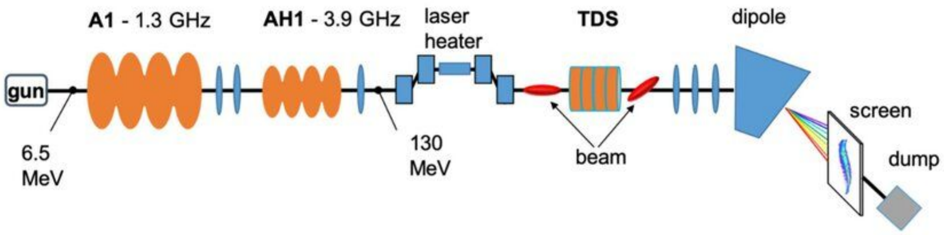
Stage 1 **Stage 2** **Stage 3**

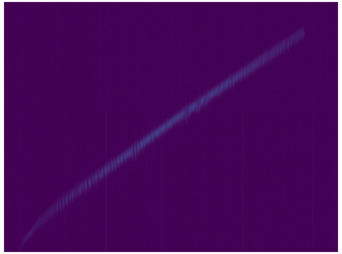
Ref: https://media.xfel.eu/XFELmediabank/catalog/Presse_XFEL_2023/r/4487/viewmode=previewview

My Project

- ❑ Beam Images are analysed in order to extract beam properties.
  - ❑ At European XFEL, the electron beam hits scintillator screens, which convert it to light, and cameras capture this light to image the beam; 60 such screens monitor the beam along the accelerator.
  - ❑ DESY operates an Image Analysis server on which the electron beam images are processed.
  - ❑ Limits of Image Analysis server – optimized to “normal” beam images, may struggle for low intensity, low charge density, faint ROI features, and TDS (Transverse Deflecting Structure) images.

- ❑ **My Objective is to:**
  - ❑ .. create a more reliable/robust method to detect the **ROI using ML methods**
    - ❑ Study of different models and performance comparisons
    - ❑ Pick the best model with highest speed and accuracy
    - ❑ Arbitrary shaped regions of interest, twin bunches





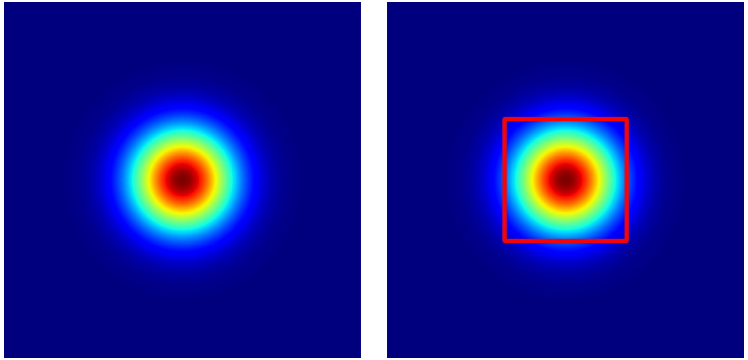
Example of beam Image

Ref:
https://media.xfel.eu/XFELmediabank/catalog/Presse_XFEL_2023/r/4487/viewmode=previewview

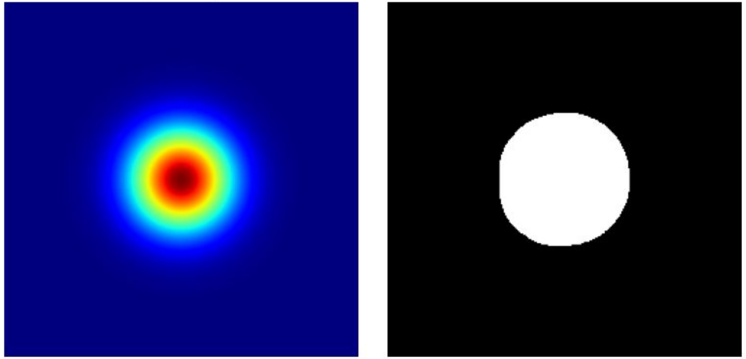
What is ROI and Why Do We Need It?

- **What is ROI?**
  Region of Interest (ROI) → the specific image area containing the electron beam signal, excluding irrelevant background.

- **Why we need ROI?**
  ROI removes noise and artifacts from electron beam images, enabling accurate beam parameter calculation; crucial for twin bunches where overlap could cause one bunch to appear in the other's calculations.



Limitation of traditional methods:
Classical algorithms usually give only a **bounding box**, which may not fully capture the fine structure of the ROI → parts of the beam signal can be missed.

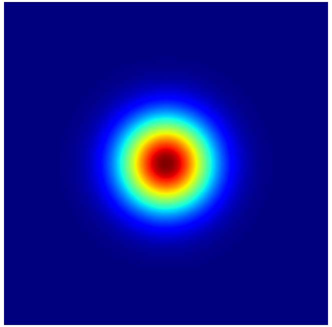


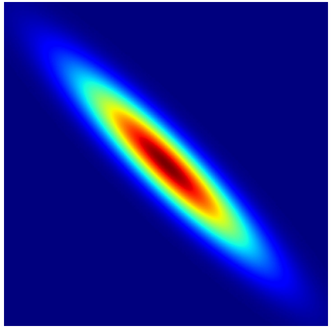
Our objective:
To obtain an **exact pixel-level mask** of the ROI, ensuring the entire beam (even faint/overlapping parts) is captured for accurate analysis.

Beam Diagnostics so far.. (Initial Approach)

Traditional Computer Vision Techniques - tried on synthetically generated images of a 2D Gaussian and real image of an electron beam.

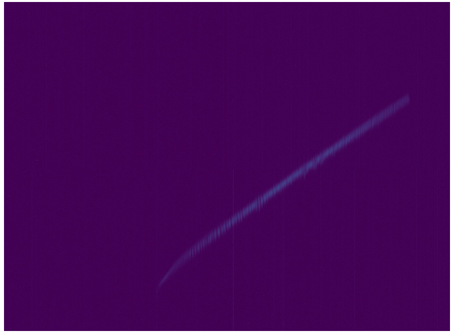
- **LoG (Laplacian of Gaussian):** Gaussian blurring followed by Laplacian.
- **Grid-based division:** Image split into blocks with intensity thresholding.
- **Otsu Thresholding + Contour Extraction:** ROI identified based on adaptive thresholding and contours.
- **Recursive Quadtree Segmentation:** Experiments with synthetic shapes of varying intensities, tuning parameters such as block size and thresholds (Otsu and fixed).
- **Bounding Box & Contour Extraction:** ROI localized using Otsu and manual thresholding, followed by bounding box placement for clearer detection.



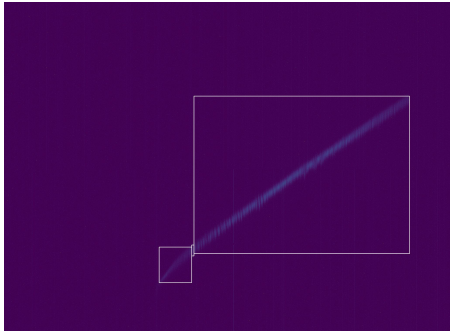


Beam Diagnostics so far.. (Initial Approach)

Traditional Computer Vision Techniques - Results on *real world images*



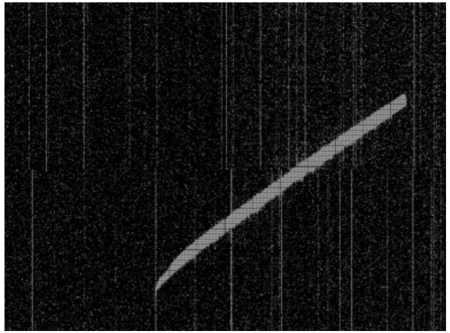
Input image



Detected ROI

The bounding box algorithm detected the primary ROI but failed to encompass the full beam extent in a single ROI

Input image



Detected ROI

The segmentation algorithm successfully identified the main beam structure but generated false positive detections in noisy background regions

The results obtained with these methods were not as good as human eye so we shifted our approach to machine learning.

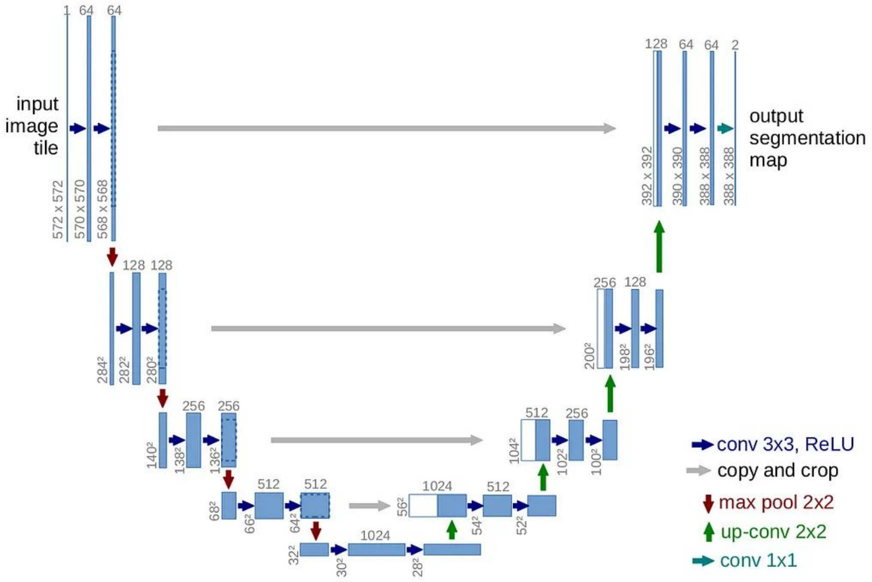
Further methods considered in this project

1. **ML Based**
    a. **YOLO** (You Only Look Once) is a real-time object detection ML method that predicts bounding boxes and class probabilities directly in a single neural network pass, enabling fast and accurate region of interest (ROI) detection.
    b. **U-Net** is a convolutional neural network designed for image segmentation, using an encoder–decoder architecture with skip connections to enable precise pixel-level ROI detection. ✅
    c. **Autoencoder** is an unsupervised neural network that compresses input into a latent space and reconstructs it, enabling feature extraction and anomaly/ROI detection through reconstruction error.

ML approaches: U-Net

Why U-Net?

- ❑ Captures both global context and fine details through its encoder–decoder structure with skip connections.
- ❑ Provides pixel-level localization → precise ROI segmentation.
- ❑ Computationally efficient
- ❑ Flexible and can be adapted → multi-class ROI
- ❑ Requires training data containing images and ground truth masks which can be generated synthetically.



U-Net Architecture

Limitations

- ❑ Requires careful tuning to avoid over/under-segmentation.
  class balancing, loss functions

Overall workflow

➡ **Generate sample: beam images (~ thousands for ML)**

with various shapes, intensities, positions, overlappings, noises

➡ **ROI Methods (conventional vs ML)**

ML: Training (80% data), Validation (20% data), Testing on real world images

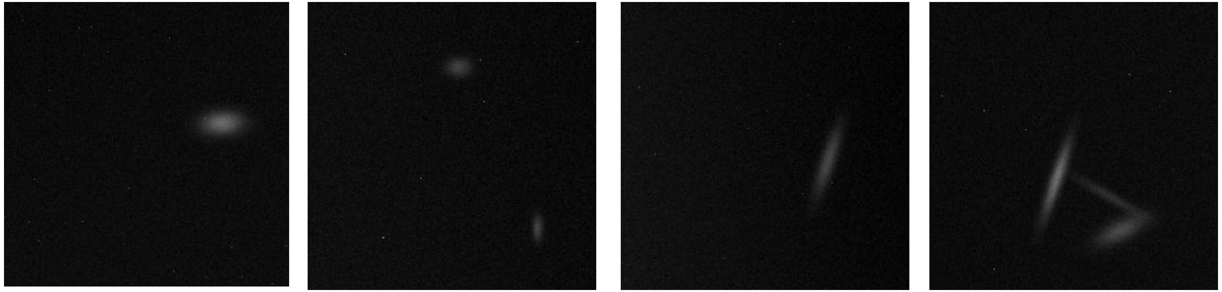
➡ **Study ROI: Input (real image) vs Output (ROI detection)**

Generate sample: beam images

U-Net: Samples

- ❑ **Images various or types (signal and background/noise)**
  - ❑ Realistic Background: Gaussian noise, gradients, readout patterns, hot pixels
  - ❑ ROI Features: Streaks, spots, arcs, diffuse blobs, rings
  - ❑ Feature Parameters: Random size, rotation, intensity, and position
  - ❑ Blending & Artifacts: Smoothing + Gaussian noise + hot/dead pixels + line artifacts
  - ❑ Faint ROI: Intensity low (0.1–0.6) → subtle, low-contrast regions
  - ❑ Augmentation: Flips, rotation, brightness/contrast, elastic & grid distortions
- ❑ **Sample Size (large dataset)**
  - ❑ **There are tens of thousands of synthetic images, with their corresponding masks**

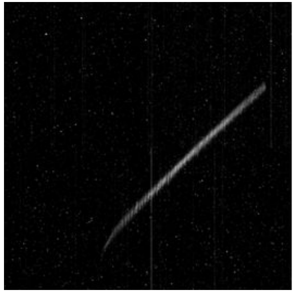
Synthetic Image Examples



ML approaches: U-Net

U-Net: Results Set-1 (V1)

- ❏ Able to identify the main ROI streak however is still considering some of the noisy regions of the image to be ROI as well.



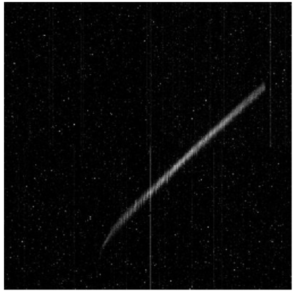
Original image

Processed mask

!!! Real images as a benchmark

U-Net: Results Set-2 (V2)

- ❏ The U-Net model is able to identify the main ROI streak well. It is not detecting any false positives.
- ❏ However, this model is conservative. It is not able to detect the entire streak.



Original image

Processed mask

ML approaches: U-Net

U-Net (Improved): Results

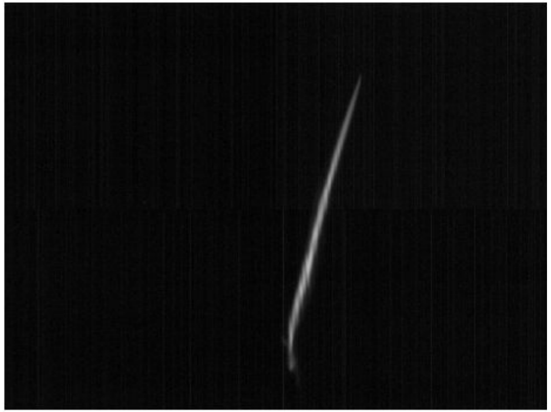
- ❑ This model shows significant improvement and is able to detect the entire beam.

The U-Net model is only trained on synthetic data.

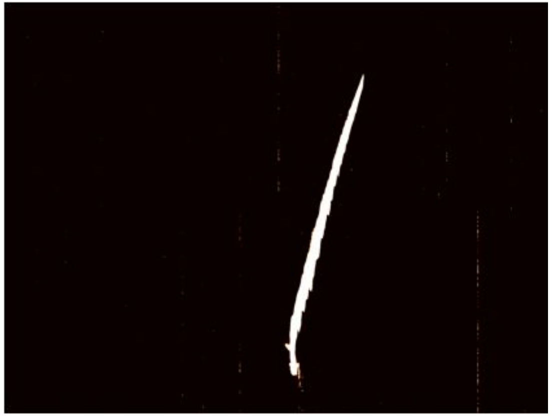
It is not overfitting. It has never seen the real data.

The model is tested on real images.

Dataset size: 30,000 images
Image size: 100x100 pixels, black and white images.



Original image



Processed mask

```
Inference completed in 0.1533 seconds
```

ML approaches: U-Net

U-Net: Model architecture (Improved vs Older versions)

Aspect	U-Net (V1)	U-Net (Improved)	Effect
Encoder filters	32 → 256	**64 → 512 (with DoubleConv + BatchNorm at each step)**	Richer features, better stability
Bottleneck	512	**1024 → 2048 (DoubleConv expansion)**	Strong latent representation for faint ROIs
Regularization	Minimal	**BatchNorm everywhere, no dropout (relies on augmentation + loss)**	Stable training, less overfitting
Decoder filters	256 → 32	**512 → 64 (skip-connections + interpolation fix)**	Sharper mask reconstruction
Dataset	Realistic background + artifacts + augmentation	Synthetic + Strong augmentation (flip, rotate, elastic, blur, brightness)	Robustness to faint/shifted ROIs
Loss	Focal + Dice	**BCE + Dice (balanced)**	Improved model focuses on structure and can capture faint regions
Result	Detects noise as ROI	Best generalization: sharp ROI detection, ignores artifacts	Best performance on real world images

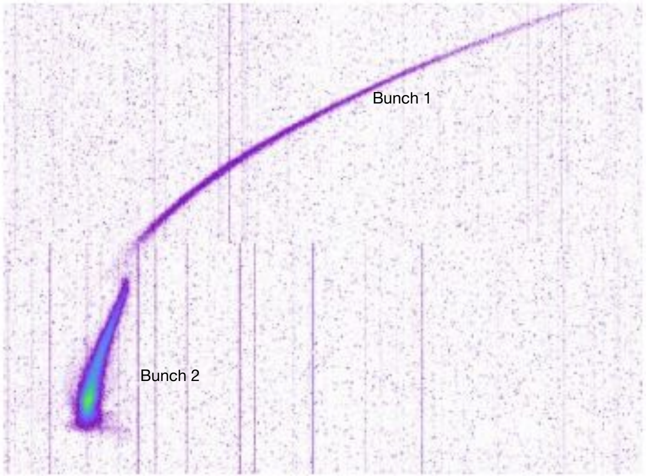
Summary and Outlook

Our Approach & Results

- ❑ Traditional methods → no improvement over existing approaches.
- ❑ ML approach → iterative improvement of **U-Net** architecture.
- ❑ Final model detects ROI in **real-world images**, trained **only on synthetic data** (handles arbitrary shapes).

Next Steps

- ❑ Extend model for **twin bunch detection**.
- ❑ **Interface model** with the actual Image Analysis server.
- ❑ **Compare performance** across different models (YOLOv5, Autoencoder).



Twin bunch
(Courtesy of Tianyun Long)

Thank you!

Prachiti Chandratreya
E-Mail: b22es030@iitj.ac.in

Extras

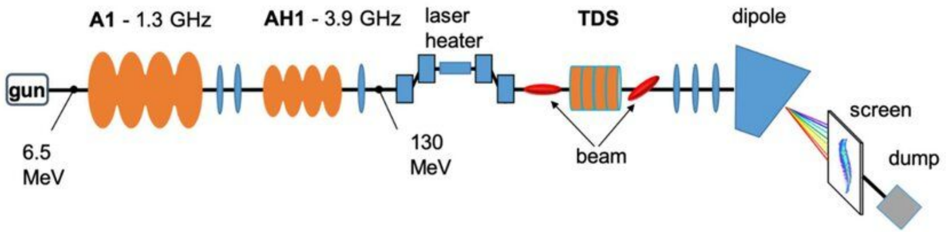
Prachiti Chandratreya
E-Mail: b22es030@iitj.ac.in

Motivation

- ❑ At European XFEL, the electron beam hits scintillator screens, which convert it to light, and cameras capture this light to image the beam; about 50 such screens monitor the beam along the accelerator.

- ❑ Images of electron beams are used to analyse parameters like emittances, energy spread, current profile which are used for measurements, to characterise the beam. Part of this analysis requires ROI (Region of Interest). It is required because we want to cut off noise and artifacts from the electron beam images. These diagnostics are also required for smooth running of the accelerator.

- ❑ Limits of Image Analysis server - optimized to "normal" beam images. May struggle for images that have low intensity, low charge density, and faint ROI features. The server is programmed to act on x and y axis so analysis of diagonal beam will not be free of noise as the coordinate system is fixed. Machine Learning will extract features independent of the camera. The server is also incapable of detecting 2 ROIs in case of multiple beams while the ML model can be trained to detect 2 features and differentiate between them by different intensity masks. If 2 beams are very close, doing it manually is very difficult so having 2 masks that fit the features is very useful.

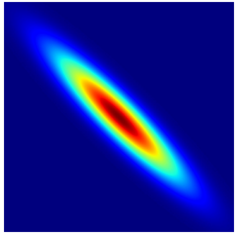


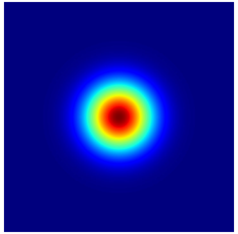
TDS IMAGE

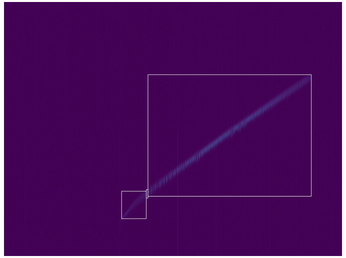
Beam Diagnostics so far.. (Initial Approach)

Traditional Computer Vision Techniques - tried on synthetically generated images of a 2D Gaussian and real image of an electron beam.

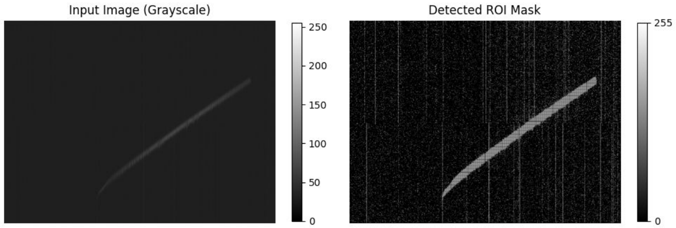
- **LoG (Laplacian of Gaussian):** Gaussian blurring followed by Laplacian.
- **Grid-based division:** Image split into blocks with intensity thresholding.
- **Otsu Thresholding + Contour Extraction:** ROI identified based on adaptive thresholding and contours.
- **Recursive Quadtree Segmentation:** Experiments with synthetic shapes of varying intensities, tuning parameters such as block size and thresholds (Otsu and fixed).
- **Bounding Box & Contour Extraction:** ROI localized using Otsu and manual thresholding, followed by bounding box placement for clearer detection.







The segmentation algorithm detected the primary ROI but fragmented the beam, and misclassified background noise



The segmentation algorithm successfully detected the ROI, but also misclassified background noise as ROI.

The results obtained with these methods were not as good as human eye so we shifted our approach to machine learning.

DESY.

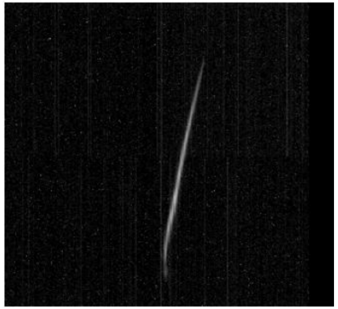
ML based methods - Comparison

Feature	**U-Net**	**YOLOv5**	**Autoencoder**
Output	**Pixel-level mask**	Bounding Box	Reconstructed image / anomaly map
ROI Precision	**Captures fine details and edges**	Detects general location, less precise edges	Highlights regions via reconstruction error, not exact mask
Global & Local Context	Encoder–decoder + skip connections	Focuses on object centers	Captures overall image patterns
Handles Faint/Small ROIs	Strong – skip connections + Dice loss preserve subtle features	Small or low-contrast ROIs may be harder to detect, but can improve with proper anchors and training	Small/faint ROIs contribute less to reconstruction error, may need post-processing
Training Data	**Images + ground truth masks**	Images + labeled boxes	Images (unsupervised possible)
Computational Cost	Moderate	Low to Moderate	Low
Best Use Case	Precise segmentation of irregular or faint ROIs	Fast detection of larger ROIs	Rough ROI localization / anomaly detection

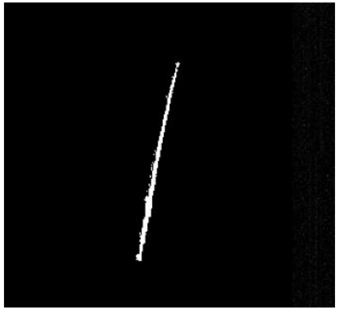
ML approaches: U-Net

U-Net: Results Set-1 (V1)

- ❏ Able to identify the main ROI streak however is still considering some of the noisy regions of the image to be ROI as well.



Original image

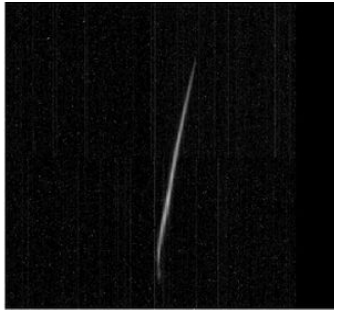


Processed mask

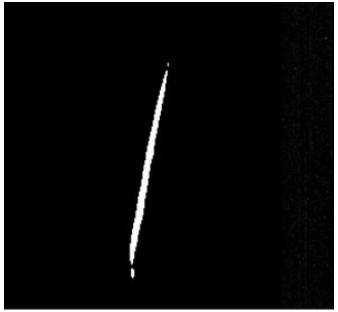
!!! Real images as a benchmark

U-Net: Results Set-2 (V2)

- ❏ The U-Net model is able to identify the main ROI streak well.
- ❏ It is not detecting any false positives. However, this model is conservative.
- ❏ It is not able to detect the entire streak.



Original image



Processed mask

ML approaches: U-Net

U-Net: Model architecture (v2 vs Old v1)

Aspect	U-Net (V1)	U-Net (V2)	Effect
Encoder filters	32 → 256	**64 → 512**	Improved model captures more complex patterns
Bottleneck	512	1024	Higher feature representation for faint ROIs
Regularization	Minimal	**BatchNorm + Dropout in encoder, bottleneck, decoder**	Reduces overfitting to noise
Decoder filters	256 → 32	512 → 64	Better reconstruction of faint features
Dataset	Simple synthetic ROIs	Realistic background + artifacts + augmentation	Improved model ignores noise and artifacts
Loss	Focal + Dice	**BCE + Dice**	Improved model focuses on structure, not pixel intensity
Result	Detects noise as ROI	Ignores noise, detects only real ROI	Better generalization

ML approaches: U-Net

U-Net (Improved): Changes in Architecture

1. Depth + skip connections keep detail

- ❑ Learns both overall structure and fine edges.
- ❑ Skip connections stop high-res details from being lost.

2. DoubleConv + BatchNorm = cleaner features

- ❑ Two convs per block capture richer patterns.
- ❑ BatchNorm keeps training stable and robust to brightness/contrast changes.

3. Smarter loss = better masks

- ❑ **BCE** gets pixel-level accuracy.
- ❑ **Dice** ensures good ROI shape/overlap.
- ❑ Together → sharp and reliable segmentations.

4. Augmentation + LR scheduling = no overfitting

- ❑ Random flips, noise, brightness shifts mimic real experiments.
- ❑ LR scheduling keeps learning smooth and avoids bad minima.

5. Checkpointing + metrics = safe training

- ❑ Monitors IoU, Precision, Recall, F1, Accuracy.
- ❑ Saves the best model automatically → no accidental regress.

This model combines deeper architecture with skip connections, double convolutions, and BatchNorm to capture fine details. Smart losses (BCE + Dice) and data augmentation ensure accurate and robust ROI segmentation. Learning rate scheduling, checkpointing, and metric monitoring make training stable and reliable.

ML approaches: U-Net

U-Net (Improved): Results

- ❏ This model shows significant improvement and is able to detect the entire beam.

The U-Net model is only trained on synthetic data.

It is not overfitting. It has never seen the real data.

The model is tested on real images.

Dataset size: 30,000 images
Image size: 100x100 pixels, black and white images.

Original image

Processed mask

```
Inference completed in 0.1840 seconds
```

DESY.

ML approaches: U-Net

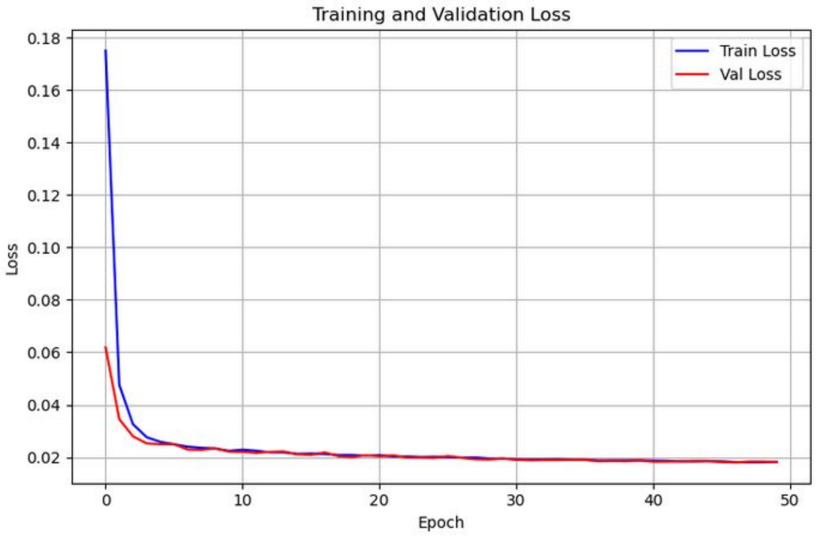
U-Net: Model architecture (Improved vs Older versions)

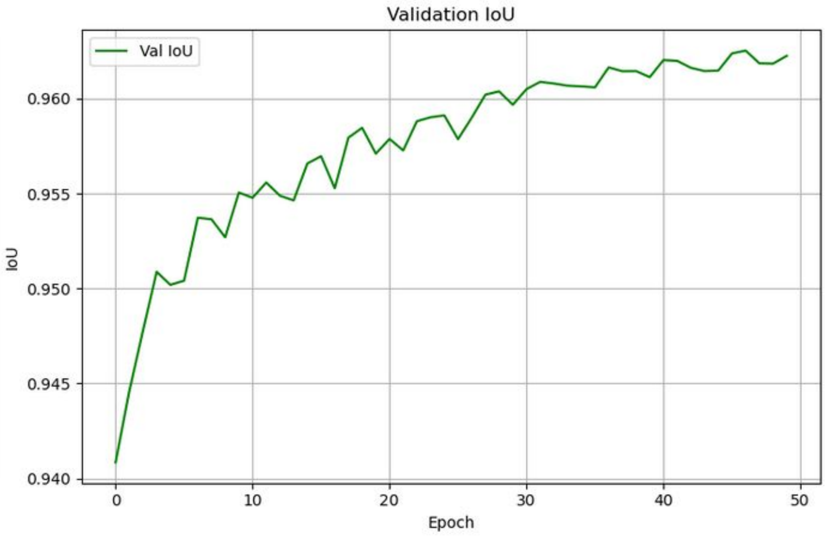
Aspect	U-Net (V2)	U-Net (Improved)	Effect
Encoder filters	64 → 512	64 → 512 (with DoubleConv + BatchNorm at each step)	Richer features, better stability
Bottleneck	1024	**1024 → 2048 (DoubleConv expansion)**	Strong latent representation for faint ROIs
Regularization	BatchNorm + Dropout in encoder, bottleneck, decoder	**BatchNorm everywhere, no dropout (relies on augmentation + loss)**	Stable training, less noise overfitting
Decoder filters	512 → 64	512 → 64 (skip-connections + interpolation fix)	Sharper mask reconstruction
Dataset	Realistic background + artifacts + augmentation	Synthetic + Strong augmentation (flip, rotate, elastic, blur, brightness)	Robustness to faint/shifted ROIs
Loss	BCE + Dice	BCE + Dice (balanced)	Improved model focuses on structure and can capture faint regions
Result	Ignores noise, detects only real ROI	Best generalization: sharp ROI detection, ignores artifacts	Best performance on real world images

ML approaches: U-Net

U-Net (Improved): Performance

- These are the plots after 50 epochs.

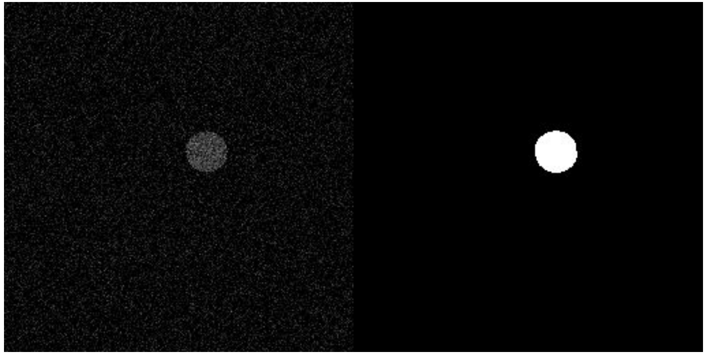




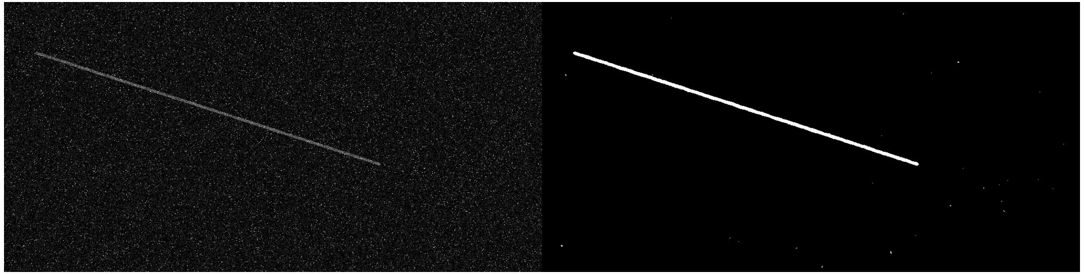
ML approaches: U-Net

U-Net (Improved): Performance on synthetic images

Input image Model prediction



Input image Model prediction

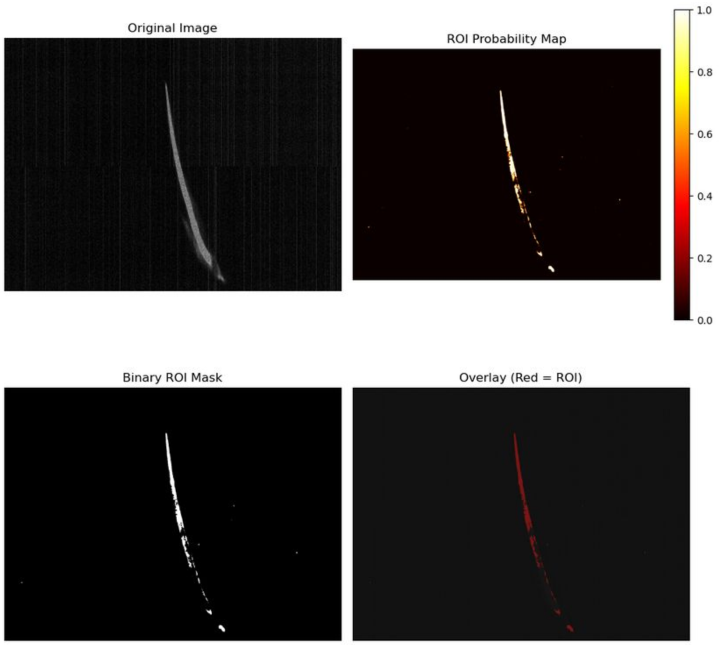


Inference on blobs, streaks and other shapes with varying levels of noise

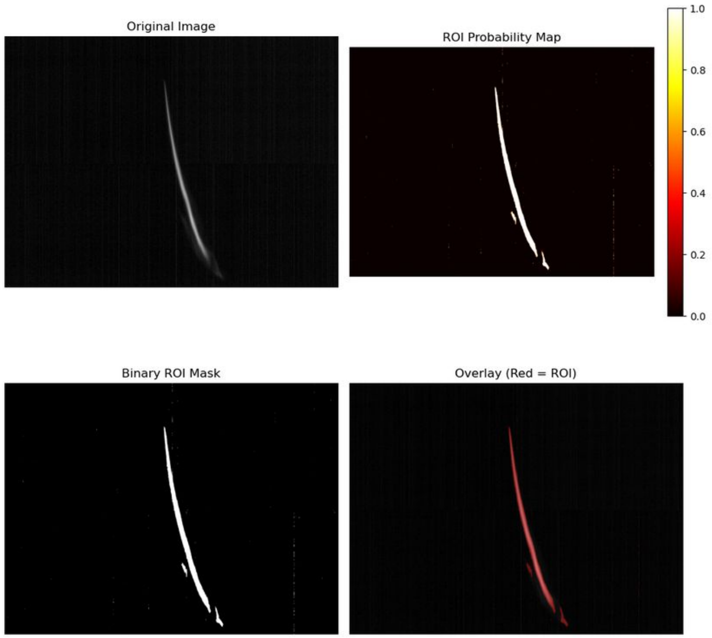
ML approaches: U-Net

U-Net (Improved-v4): Performance on real-world images

Inference on colormap of real image



Inference on real image enhanced by gamma function



DESY.