

Full-chip Voltage Contrast Inference Using Deep Learning You Only Look Once: Voltage Contrast (YOLO-VC)

Kelvin Yih-Yuh Doong^a, ChenPo Lin^b, Sheng-Che Lin^b

^aPDF Solutions Inc., Santa Clara, California, USA

^bPDF Solutions Inc., Zhubei City, Hsinchu County, Taiwan

ABSTRACT

The electron beam inspection methodology for voltage contrast (VC) defects has been widely adopted in the early stages of sub-10nm logic and memory technology development, as well as in new product introductions. However, due to throughput limitations, full-chip inspection at the 300mm wafer scale remains impractical for yield ramp and production applications. To address this challenge, we propose a deep-learning approach for full-chip voltage contrast inference. By modifying and enhancing the You Only Look Once (YOLOv7) model into YOLO-Voltage Contrast (YOLO-VC)—where YOLOv7 is the most efficient object detection neural network—the voltage contrast of metal patterns across the entire chip can be accurately predicted. By mapping the voltage contrast response at the full-chip level, the inspection recipe can be optimized to focus on critical care areas where defects are most likely to occur. We present the methodology, including process flow, image-to-image registration, gray-level classification, model training and validation, and a performance benchmark comparing YOLOv7 and YOLO-VC. Finally, we propose leveraging the full-chip VC density map for area of interest (AOI) selection to optimize throughput and enhance the capture rate of VC defects.

Keywords: electron beam inspection, voltage contrast, YOLO-VoltageContrast (YOLO-VC), object detection neural network

1. INTRODUCTION

Electron beam inspection (EBI) has outperformed optical inspection in detecting smaller physical defects, electrical defects, voltage contrast (VC) defects, and abnormalities in pattern printing fidelity [1]. While EBI offers impressive advantages, its significantly slower scanning throughput—caused by pixel-by-pixel scanning at sub-10nm resolution—limits its widespread use as a production inspection tool.

To address this limitation, electron beam tools utilize vectorized scanning to selectively inspect areas within a large field of view (FOV) of $180 \times 180 \mu\text{m}^2$ without moving the stage [2]. This approach improves throughput by reducing unnecessary stage movement, allowing large FOV scanning without wasting resources on unimportant regions. Additionally, multi-beam inspection (MBI) systems incorporate multiple beamlets, high-speed stages, and high-speed computational architectures to enable parallel inspection, significantly accelerating data collection [3]. DirectScan technology enhances this further by performing sophisticated product layout analysis to determine VC-detectable locations within the design, guiding electron beam vectorized landings to those locations. With a stage speed of 100mm/sec and an FOV of $45 \times 45 \mu\text{m}^2$, it enables full wafer scans within 2 to 4 hours [4].

VC inspection has been extensively studied in both memory and logic technologies, with optimizations focusing on special process flows, scan direction, multiple scan passes per pixel, and design-guided scanning. For instance, a tailored process optimizes the detection of VC defects arising from deep trench etching in embedded dynamic random-access memory (eDRAM) [5]. By adjusting the scan direction and modifying the number of electron signal collection passes per pixel, SRAM-specific open/short failures in the middle-of-line process can be detected using VC in positive mode [6]. Additionally, vector scan-enabled e-beam systems selectively charge multiple word-line (WL) transistors in NAND flash memory. By in-situ charging the WL transistors, these systems transition the transistors from a floating state to an on-state, enabling VC-based open detection of deep contacts in NAND flash memory [7].

However, VC detection in logic devices is more prone to failure due to nuisance signals arising from complex connectivity between capacitors, resistors, transistors, and long interconnect lines. To mitigate this, a classification methodology [8] based on net tracing is employed, categorizing VC responses into three groups:

- Net traces connected to the active layer
- Net traces connected to the gate
- Floating metal

Traditional VC inspection methods rely on either the uniformity and consistency of VC signals in memory cells or rule-based models linking VC signals to net connections to minimize nuisance rates. However, in logic products, VC complexity increases due to variations in transistor connection topologies. Figure 1 illustrates how transistor connection topology impacts VC. As the number of stacked NMOS transistors increases, the VC response at the drain transitions from bright to dark. Stacked NMOS structures are commonly found in multi-input static standard cell gates, such as NAND, AND-OR-INVERTER, XOR, clocked AND, and clocked NAND gates.

To address this challenge, we propose a deep learning-based neural network that learns VC responses based on various transistor connections and subsequently infers the VC of layout patterns at a full-chip scale. This allows VC inspection recipes to be optimized for the densest areas of interest (AOI), increasing defect capture rates and overall inspection efficiency.

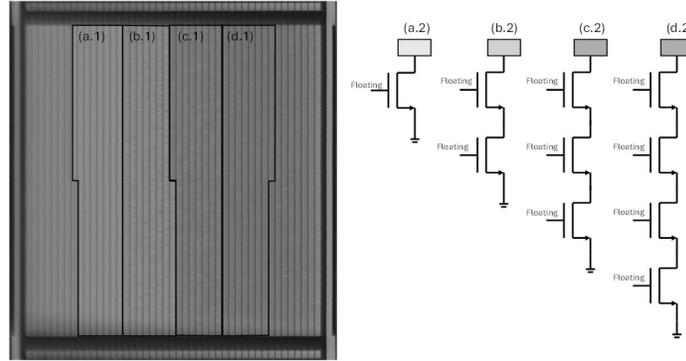


Figure 1: SEM micrograph of voltage contrast inspection. (a.1) ~ (d.1) are images of NMOS with Drain connected to Metal-0 as inspection layer, Source connected to ground and Gate as floating. (a.2) ~ (d.2) are schematics of (a.1) ~ (d.1), respectively.

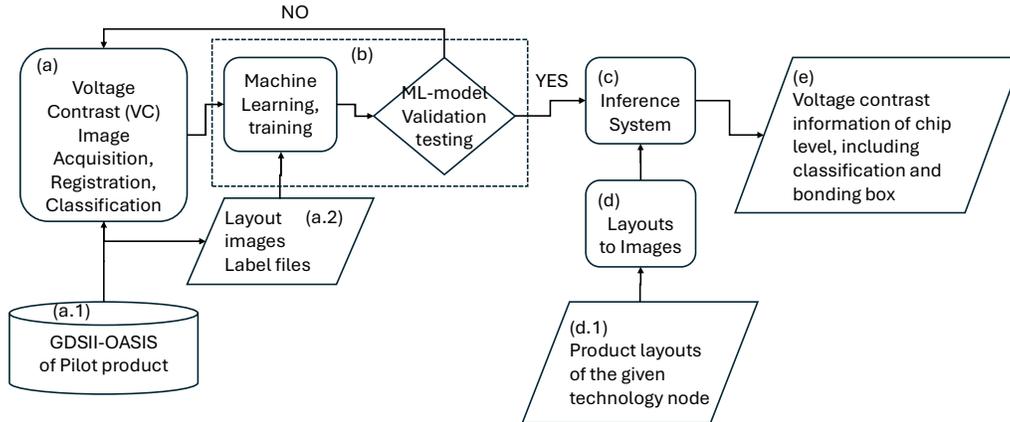


Figure 2: The flow chart of machine-learning enabled voltage contrast (VC) image system. (a) VC acquisition, registration, gray level quantization and label onto layout (GDS-II/OASIS). (a.1) input of (a) which is product physical layout (GDS-II/OASIS). (a.2) output of (a) which includes images (RGB-formatted) from layout and labels associated to the given image. (b) a machine-learning engine used for training, validation, and testing. (c) VC inference system. (d) software used to slice the whole product layout into millions of small layouts and transformed into images of 640x640 pixels. (d.1) the product layouts of the given technology modeled in (b). (e) VC information of the full chip including classifications and bonding boxes.

2. METHODOLOGY

The process flow illustrated in Figure 2 consists of the following components:

- (a) A voltage contrast (VC) image acquisition system
- (b) A deep learning engine for model building through training, validation, and testing
- (c) An inference system that provides VC classifications and bounding boxes
- (d) A program for transforming layouts (GDS-II/OASIS) into RGB images

(e) An output system that generates text-format results from the inference system, containing VC classifications and corresponding bounding boxes

A pilot product layout (a.1) is first split and fed into component (a), where a layout file of several gigabytes is sliced into smaller sections, each stored as a megabyte-sized layout. These smaller layouts focus on the specific areas inspected by the scanning electron microscope (SEM) to acquire VC images. Each layout contains only the inspected layer, which is used for registration and alignment with the VC images. By registering the VC image onto the corresponding inspected layer layout, the gray levels of objects—either rectangles or polygons (in advanced nodes, most are rectangles)—can be quantified into various classes (four classes are used in this case). By combining the registered coordinates, bounding boxes, and class labels, the system generates a text file that serves as a label file for machine learning applications.

In Figure 2(b) and 2(c), a modified version of "You Only Look Once" (YOLOv7) [9] is utilized as the deep learning engine for model building through training, validation, and testing. This modified version is referred to as YOLO-VoltageContrast (YOLO-VC). YOLO is an object detection algorithm that processes an image in a single forward pass through a neural network, outputting both classifications and bounding boxes for detected objects. Since YOLO is designed to handle object recognition in images with three- or four-color channels, this work enhances the image loader in YOLOv7 by transforming multiple layers into a matrix of size $M \times N \times Z$, where Z ranges from 3 to 512, and M/N is set to 640 to balance training/testing speed and accuracy.

In Figure 2(d) and 2(e), customized tools are developed to process millions of $640 \times 640 \times Z$ images as inputs for full-chip VC inference. The system outputs the corresponding classification results and bounding boxes. Given the total chip size of 100 mm^2 , the number of images is estimated based on an image size of $1.28 \times 1.28 \mu\text{m}^2$, assuming a 2 nm pixel resolution. This results in approximately 61 million images (~61M) derived from the full-chip division by individual image sizes.

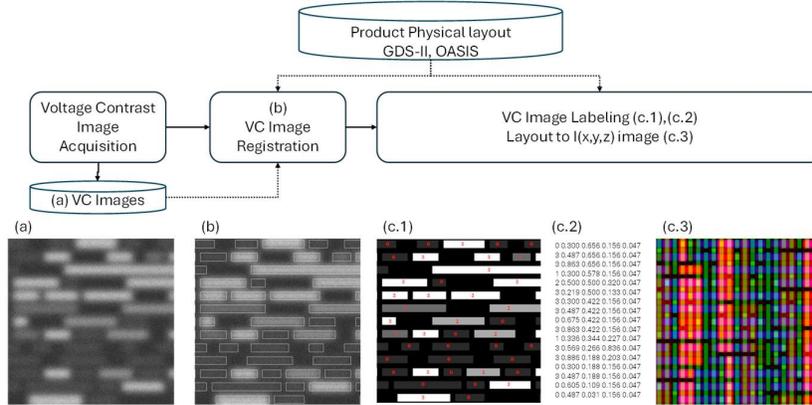


Figure 3: The flow of voltage contrast image acquisition, registration, gray level quantization and label onto layout (GDS-II/OASIS). (a) VC image. (b) VC image registered onto the inspection layer (M1 used here). (c.1) VC gray level is quantified and labeled. (c.2) classifications of VC image are at column-1, the X-/Y-coordinates of the bonding box center in column-2/-3. (c.3) image transformed from layout (GDS-II/OASIS).

A. Dataset building

Figure 3 describes the operation in Figure 2(a). A Python script is programmed to parse the layout data and convert into RGB-formatted images. Since the layout contains more than three layers—exceeding the representation limits of standard image formats like RGB or CMYK—a three-dimensional matrix data structure is employed to store information parsed from multiple layers of the layout. In this matrix, the first and second dimensions correspond to the X- and Y-coordinates, while the third-dimension stores individual layers from the first process layer up to the inspection layer. Each object within a specific layer is encoded as a binary image, where:

$$l(x, y, z)=1, \text{ if } (x, y, z) \text{ is inside or on the edge of an object at the } z\text{-th layer}$$

$$l(x, y, z)=0, \text{ otherwise}$$

Here, z ranges from 0 to $N-1$, where N represents the total number of layers electrically connected to the inspection layer from the first layer. The VC image shown in Figure 3(a) is acquired in either positive or negative VC mode and is mathematically represented as:

$$vc(x, y, z)$$

where x and y index the X-, Y-coordinates within the range $0 \leq x \leq m-1$ and $0 \leq y \leq n-1$.

In Figure 3(b), the VC image $vc(x,y,z)$ is registered onto the corresponding layout representation $l(x,y,z)$. The registration process is solved by finding the maximum of matrix $L(x,y,z)$ using the following method:

$$L(x, y, z) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} l\left(x - \frac{m}{2} + i, y - \frac{n}{2} + j, z\right) * vc(i, j, z),$$

where $\frac{m}{2} \leq x \leq M - 1 - \frac{m}{2}$, and $\frac{n}{2} \leq y \leq N - 1 - \frac{n}{2}$

$$\max L(x, y, z)$$

Once $vc(i,j,z)$ is registered onto $l(i,j,z)$, the layout objects within $vc(i,j,z)$ are classified based on either the mean or median of the VC values. The classification result is then assigned to the corresponding object, as shown in Figure 3(c.1).

A Python script is used to parse the classification data from Figure 3(c.1) into a text file (Figure 3(c.2)), which serves as the label dataset for training, validation, and testing.

All layout layers, including both the inspected and manufactured layers, are transformed into image-formatted datasets. For example, the first, second, and third layers are mapped to the RGB channels of Image-0 in Figure 3(c.3). The process repeats until all N layers are processed, resulting in a total image count of:

$$[N/3]+1$$

Figure 3(c.2) and 3(c.3) represent the dataset used for training. The entire workflow is implemented in software to minimize errors caused by manual labeling. The labeling process involves:

1. Drawing bounding boxes (as shown in Figure 3(b))
2. Assigning class labels (as depicted in Figure 3(c.1))

B. Object detection neural network: YOLO-Voltage Contrast (YOLO-VC)

Figure 4 presents the proposed YOLO-VC model. YOLOv7 was originally designed to detect images in an industry-standard format, where the maximum number of channels is limited to four. However, with the increasing complexity of advanced technologies, the number of process layers has grown to hundreds, including both physical and marker layers. As shown in Figure 4(a), YOLOv7 must be modified to accommodate multi-channel images. The multi-channel image loader takes all available layers as inputs, spanning from the initial process step to the inspected layer, enabling deep learning models to process the complicated connections among layers.

The backbone network, depicted in Figure 4(b), consists of a convolutional neural network, batch normalization, and activation functions, as detailed in [9]. The meaningful features are captured in a hierarchical way across multiple scales. The neck network, illustrated in Figure 4(c), extracts feature from the backbone network across three grid sizes. It accumulates and filters these features, enhancing both spatial and semantic information at different scales. The head network, shown in Figure 4(d), is responsible for predicting object confidence scores, categories, and anchor frames. Finally, as shown in Figure 4(e), the output is generated in text files that contain information about the classified objects, including their categories, locations, and bounding boxes.

Each grid cell contains three anchors, with each anchor predicting the following parameters:

- x,y : The center of the bounding box, relative to the grid cell
- w, h : The width and height of the bounding box, scaled to the whole image
- Confidence score: The Intersection over Union (IoU) between the predicted and ground truth bounding box

Including four classes, the final predictions are of size: 20x20x27, 40x40x27 and 80x80x27.

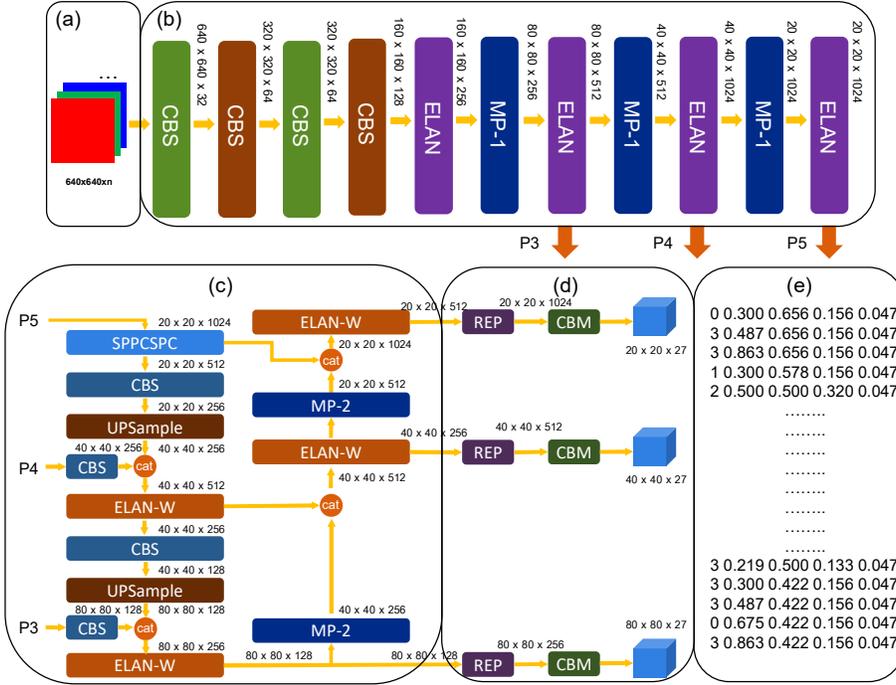


Figure 4: Structure diagram of the YOLOv7-voltage contrast (VC) model. (a) multi-channels image loader. (b) backbone. (c) neck. (d) head (e) results.

3. DATASET PREPARATION, TRAINING AND VALIDATION

All experiments are conducted on Nvidia RTX-4080 (16GB) GPUs. The training and validation time range from 10 to 40 hours. Since GPU is installed on a server with multiple jobs loaded, the runtime can be further improved as needed. Two benchmark datasets are used for experiment comparison between YOLOv7 and YOLO-VC. One dataset corresponds to the inspection layer set at Metal-1 while the other is at Metal-2. The Metal-1 dataset consists of nine layers, whereas the Metal-2 dataset includes eleven layers. The layers are named as the following:

•NAA: n-type active	•PAA: p-type active	•GATE: channel gate
•SD: source/drain epi-metal	•SD_CO: contact to source/drain	•G_CO: contact to gate
•M0: first metal	•V0: via to M0	•M1: second metal
•V1: via to M1	•M2: third metal	

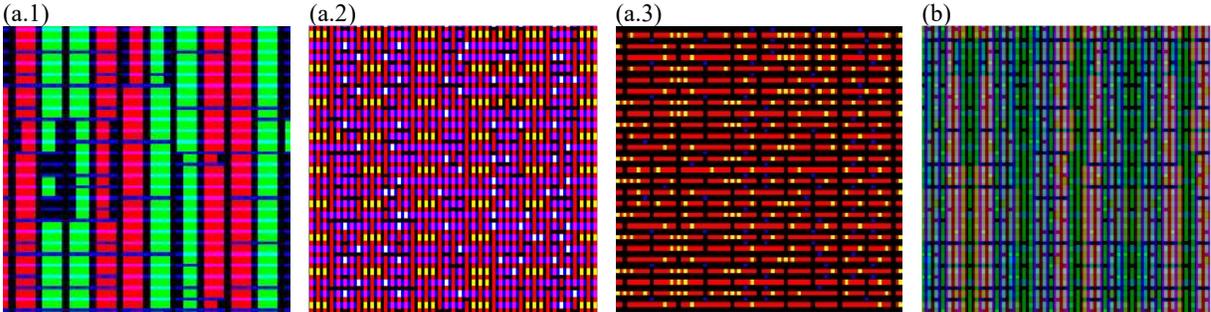


Figure 5: Images used in YOLO-VC and YOLOv7. (a.1) Image of NAA, PAA and GATE (a.2) Image of SD, SD_CO and G_CO (a.3) Image of M0, V0 and M1. (a.1), (a.2) and (a.3) are used for YOLO-VC. (b) Image of all layers used for YOLOv7.

Since YOLOv7 can only process RGB-formatted images, layers are mapped to the two to six significant bits of the RGB format in the YOLOv7 dataset as shown in Figure 5(b). For YOLO-VC, every three layers are combined into a single image, resulting in three or four images paired with a corresponding label file as illustrated in Figure 5(a1.-3). Table 1 summarizes the layer mapping used in both cases.

Table 1: Summary of image RGB and layer map versus the benchmark.

		NAA	PAA	GATE	SD	SD_CO	G_CO	M0	V0	M1	V1	M2	
YOLO-VC	Image Serial	Image0			Image1			Image2			Image3		
	Color	R	G	B	R	G	B	R	G	B	R	G	B
	M2	255	255	255	255	255	255	255	255	255	255	255	0
	M1	255	255	255	255	255	255	255	255	255	0	0	0
YOLOv7	Image Serial	Image0											
	Color	R	R	G	G	G	G	G	B	B	B	B	B
	M2	64	128	8	16	32	64	128	16	32	64	128	0
	M1	64	128	8	16	32	64	128	16	32	0	0	0

The label file categorizes data into four classes: **F**, **FGATE**, **FNAA**, and **GND**.

F: If all layouts connected to the inspected target are connected only to the metal or via layers, they are labeled as **F**.

FGATE: If all layouts connected to the inspected target are connected only to the metal, via, or gate layers, they are labeled as **FGATE**.

FNAA: If any layout connected to the inspected target is connected to a floating **NAA** (n-type active) without a connection to **GND**, it is labeled as **FNAA**.

GND: If any layout connected to the inspected target is connected to **GND**, it is labeled as **GND**.

Table 2 provides an overview of the total number of images and labels. All images are derived from the same design, with the layout transformed according to the flow in Figure 3, and the grid resolution set to two nanometers per pixel.

Table 2: Summary of datasets

Counts in total	Metal-1		Metal-2	
	Image +Label	Labels	Image +Label	Labels
YOLO-VC	6520	1010340	7315	298025
YOLOv7	6760	1041925	7540	313705

Table 3: Comparison table of YOLOv7 and YOLO-VC

Model		YOLO-VC				YOLOv7			
Inspection layer	Class	P	R	mAP@.5	mAP@.5:.95	P	R	mAP@.5	mAP@.5:.95
Metal-1	all	0.998	0.998	0.997	0.996	0.99	0.99	0.997	0.967
	F	1	0.999	0.999	0.997	1	0.996	0.998	0.962
	FGate	0.996	0.996	0.997	0.996	0.974	0.985	0.995	0.963
	FNAA	0.999	1	0.996	0.996	0.999	0.993	0.997	0.98
	GND	0.998	0.997	0.997	0.996	0.989	0.989	0.997	0.965
Metal-2	all	0.966	0.97	0.984	0.979	0.851	0.816	0.836	0.818
	F	0.995	0.992	0.998	0.994	0.994	0.948	0.975	0.958
	FGate	0.924	0.953	0.97	0.966	0.701	0.886	0.813	0.797
	FNAA	0.999	0.998	0.995	0.988	0.971	0.721	0.741	0.72
	GND	0.948	0.937	0.973	0.969	0.739	0.71	0.813	0.799

4. RESULTS AND DISCUSSION

The validation results of YOLO-VC and YOLOv7 are depicted in Table 3, which compares precision, recall, mean average precision, at 50% and 95% of intersection over union. Precision and Recall are defined as the following:

$$\text{Precision} = \frac{TP}{(TP+FP)}, \text{Recall} = \frac{TP}{(TP+FN)}$$

where TP: true positive, FP: false positive, FN: false negative. By comparing the validations of the inspection layer on metal-1 and metal-2, YOLO-VC are more accurate than YOLOv7 on both precision and recall (P/R M1: 1.0/1.0 vs 0.99/0.99; M2: 0.97/1.0 vs 0.85/0.82). In metal-1 dataset, YOLO-VC is 1% better than YOLOv7 in the four classes. In metal-2 dataset, YOLO-VC outperforms YOLOv7 5%~20% in the four classes. For F-class, its voltage contrast is only involved with metal and via layer, both networks almost have the same performance. For FGATE, FNAA and GND, VC response is more complicated than the connection among metal and via layers. It involves all layers connected to the inspected layout pattern. With the aid of multi-channel input in YOLO-VC, the network can be trained through all of channels (layers) and the kernel can be tuned with the correlation among multiple channels. Figure 6 shows the output from the first stage of convolutional network. The network has learned a variety of layer-to-layer connections. The value of thirty-two kernels can be treated as the connection strength among layers. From (a.1) to (a.19), it shows the kernels learned by NAA, PAA and GATE dominated connections. From (a.20) to (a.32), the kernels focus on the metal and via connections.

With the help of YOLO-VC, voltage contrast (VC) inference at the chip level can be efficiently achieved by dividing the entire product design into millions of small layout segments. These layouts are analyzed using YOLO-VC to determine the voltage contrast response for each inspection layout pattern, along with their respective locations and classifications. Figure 7 illustrates the voltage contrast density map for the class-GND.

Two specific areas, each measuring $140 \times 140 \mu\text{m}^2$, are selected for this study: one represents a logic design-intensive region, and the other is a combined area of SRAM and logic design. For each area, twelve thousand image sets, along with their corresponding label files, are processed through the YOLO-VC inference flow. The inference time per image averages 190 milliseconds, demonstrating the efficiency of YOLO-VC for large-scale chip inspection tasks. This timing was achieved using an NVIDIA RTX-4080 SUPER GPU, highlighting its capability to process extensive datasets in a reasonable timeframe. The speed can be further improved by using the newest hardware. The results are then remapped into a unified density map, as shown in Figure 7. These findings highlight that the voltage contrast response varies significantly based on both the design content and the specific location within the chip.

The capability of YOLO-VC to map out the voltage contrast response at the full chip level offers significant potential for optimizing ebeam VC inspection. By identifying areas with meaningful voltage contrast variations, inspection efforts can focus on regions that are most likely to yield valuable insights, reducing non-effective inspections. For example, in cases where a circuit failure results in an open circuit, the VC of the inspected pattern would appear bright under normal conditions but would turn dark in the event of failure (or vice versa, depending on the specific failure mode).

Figure 8 proposes a deep learning neural network architecture designed to learn the VC response based on various transistor connections. This network aims to infer the VC across the full chip scale, enabling a more comprehensive and predictive approach to VC mapping. As a result, VC inspection recipes can be refined to prioritize the densest areas of interest, enhancing the capture rate and overall inspection efficiency. This optimization not only improves defect detection but also reduces inspection costs and time, making the process more effective for large-scale chip manufacturing.

5. CONCLUSION AND FUTURE WORK

In this work, we implement an enhanced architecture of YOLO-VC for voltage contrast (VC) analysis, incorporating multi-channel image capability. Compared to YOLOv7, this new architecture outperforms by 5% to 20% on Metal-2 and 1% to 2% on Metal-1 in terms of precision and recall. With the help of YOLO-VC, chip-level VC inference is demonstrated in two specific areas, refining the priority of inspection area selection and enabling the inspection recipe to be aware of the density distribution of specific voltage contrast responses.

Future studies should focus on improving inference speed and analyzing the root causes of classification failures to further enhance the robustness and efficiency of this methodology. In this work, the inference speed reaches only five frames per second (FPS), which is the lower bound of YOLOv7. Previous reports indicate that YOLOv7 can achieve between 5 and 190 FPS.

During the classification failure analysis, we observed that most misclassified patterns contained long interconnections extending beyond the image boundaries. This lack of complete connection data within the input image contributes to classification failures. While increasing image size could potentially resolve this issue, it would further reduce inference speed, presenting a trade-off between accuracy and efficiency.

ACKNOWLEDGEMENTS

Chiao-Jo Tung, a senior student in the Biomechanics Engineering Department at National Taiwan University, contributed initial Python scripts for her capstone project on parasitoid wasp species counting and sex recognition. The scripts are used for image processing tasks, including transforming databases into images, image registration, and labeling. These scripts were developed to explore the feasibility of using YOLOv7 for the project.

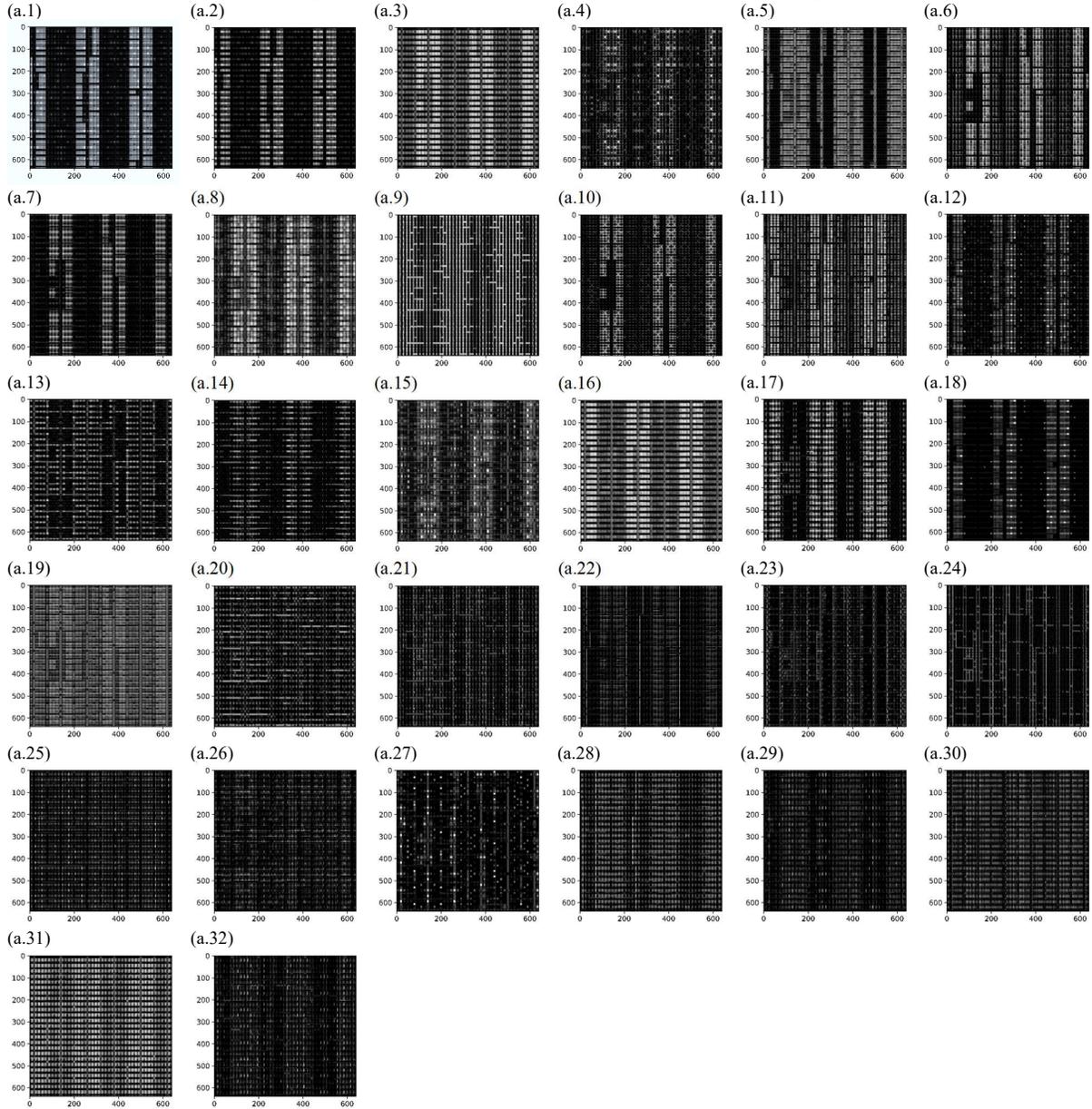


Figure 6: (a.1) ~ (a.32) Images output from the 1st convolutional neural network of *Error! Reference source not found.* (b).

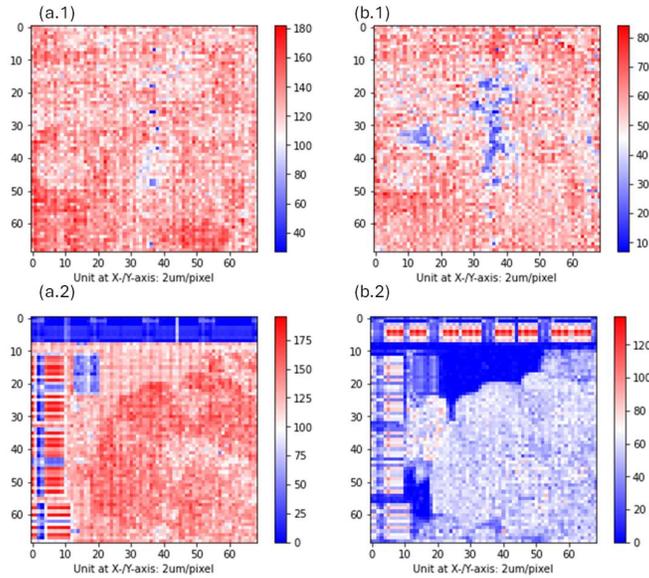


Figure 7: Voltage contrast density map of the area of interest, where the color indicates the counts of GND. (a.1) and (b.1) illustrate areas with intensive logic circuits, with the inspection layers being M1 and M2, respectively. (a.2) and (b.2) show areas combining logic circuits and SRAM cache, with the inspection layers also being M1 and M2, respectively.

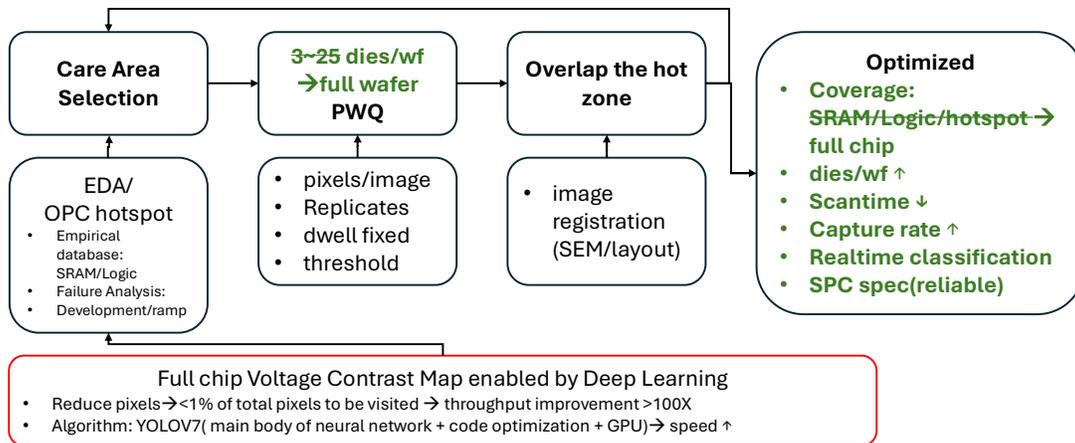


Figure 8: VC inspection recipe set up flow guided with full chip VC map.

REFERENCES

- [1] Ankush Oberai, Jiann-Shiun Yuan, “Smart E-Beam for Defect Identification & Analysis in the Nanoscale Technology Nodes: Technical Perspectives”, *Electronics* **2017**, 6(4), 87, doi: 10.3390/electronics6040087
- [2] Oliver D. Patterson, Hsiao-Chi Peng, Haokun Hu, Chih-Chung Huang, and Panneer S. Venkatachalam, “Creative Use of Vector Scan for Efficient SRAM Inspection”, 2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), doi: 10.1109/ASMC49169.2020.9185264
- [3] Eric Ma, Weiming Ren, Xinan Luo, Shuo Zhao, Xuerang Hu, Xuedong Liu, Chiyan Kuan, Kevin Chou, Martijn Maassen, Weihua Yin, Aiden Chen, Niladri Sen, Martin Ebert, Lei Liu, Fei Wang, and Oliver D. Patterson "Multi-beam Inspection (MBI) development progress and applications", *Proc SPIE 11325, Metrology, Inspection, and Process Control for Microlithography XXXIV*, 113250F (2020)
- [4] Andrzej J. Strojwas, Tomasz Brozek, Kelvin Doong, Indranil De, Xumin (William) Shen, and Marcin Strojwas, “Novel E-beam Techniques for Inspection and Monitoring”, 2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), doi: 10.1109/EDTM53872.2022.9798308

- [5] Oliver D. Patterson; Xing J. Zhou; Rohit S. Takalkar; Katherine V. Hawkins; Eric H. Beckmann; Brian W. Messenger, "Methodology for trench capacitor etch optimization using voltage contrast inspection and special processing", IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC), 2010, pp.109-114, doi: 10.1109/ASMC.2010.5551433
- [6] Hao-Yu Chien, Chan-Hao Hsu, Yue-Ying Yen, and Tzung-Hua Ying, "A Case Study on Inline Defect Diagnosis by Applying E-beam Inspection System", 27th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), May 2016, pp.285-288, doi: 10.1109/ASMC.2016.7491151
- [7] Muneyuki Fukuda, Kazuhisa Hasumi, Takashi Nobuhara, Hirohiko Kitsuki, Zhigang Wang, Kazuhiro Nojima, Yusaku Suzuki, Akira Hamaguchi, Masashi Kubo, and Masaya Hosokawa, "In situ electrical property quantification of memory devices by modulated electron microscopy", J. Micro/Nanopattern. Mater. Metrol. 041605-1 Oct-Dec 2023, Vol. 22(4)
- [8] Weihong Gao, Xuefeng Zeng, Peter Lin, Yan Pan, Ho Young Song, Hoang Nguyen, Na Cai, Zhijin Chen, Khurram Zafar, "Net tracing and classification analysis on E-beam die-to-database inspection," Proc. SPIE 9778, Metrology, Inspection, and Process Control for Microlithography XXX, 97783Q (25 March 2016); doi: 10.1117/12.2235347
- [9] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Nov. 2023, pp. 7464-7475