Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 02, PP. 187-201, 2025
DOI: https://doi.org/10.54216/JISIoT.160214

SCNN-UNet: A Novel Deep Learning Approach for Pulmonary Embolism Detection in COVID-19 Patients Using Super Pixel Segmentation

Sukhwinder Bir[1,*], Vijay Dhir[2]

[1]Research Scholar, Sant Baba Bhag Singh University, Punjab

[2]Professor, Department of Computer Science and Engineering, Sant Baba Bhag Singh University, Punjab

Emails: sukhwinderbir@gmail.com; dr.vijaydhir@ gmail.com

Inventory management is crucial for optimizing consumer demand and supply chains in e-commerce companies. This is the stage at which precise inventory forecasting becomes necessary for forecasting future demand patterns and stock levels. Traditional forecasting methods often struggle with e-commerce data due to seasonality, sudden changes in customer behavior, and nonlinearity. Machine learning (ML) and deep learning (DL) techniques have become powerful weapons for inventory prediction because they can analyze huge amounts of data with high dimensionality. E-commerce firms can improve their resource allocation, inventory management, and customer experience in highly competitive market environments. This paper proposes different types of inventory forecasting models and mainly evaluates the applicability of sophisticated machine learning algorithms. While we commonly use old methods like Random Forest, ARIMA, and MLPs, they often lack the necessary robustness to nonlinearity within inventory data. To address these problems, we introduce a novel method that combines convolutional neural networks (CNN) and XGBoost called CNN-XGBoost, which provides better feature extraction than the conventional prediction model and regression performance. We then compared CNN-XGBoost's performance to traditional forecasting methods (another common approach to contextualizing predictive model performance) using a 52-week simulated dataset in which we mimic patient data growing over time. We used key performance metrics such as R2, mean squared error (MSE), and mean absolute percentage error (MAPE) to assess each model's accuracy. The CNN-XGBoost model performed much better than others, with an R2 of 0.78, which means our proposed model can explain more variation compared to other competitors, as depicted in the results section. It also had the best MSE of 0.15, indicating better predictive performance. The CNN-XGBoost model demonstrated promising prospects as a robust inventory forecasting tool for commerce despite its slightly higher MAPE value (0.69), suggesting some vulnerability to outlier data points. This study demonstrates the potential of using a convolutional neural network in combination with gradient boosting techniques to tackle the complexity of stock management issues and the fact that it outperforms based line methods by a large margin.

Received: January 25, 2025 Revised: February 20, 2025 Accepted: March 15, 2025

Keywords: Covid-19; Prediction; Deep Learning

1. Introduction

E-commerce inventory forecasting is the task of predicting the future sales of a product an e-commerce retailer has using statistical methods and machine learning algorithms. This involves analyzing past sales and considering other factors such as promotions, events, and market trends to determine the correct stock level for every individual item to satisfy demand projections. E-commerce inventory forecasting seeks to maintain an optimal level of in-stock inventory that can fulfill the demand without overstocking, which may lock your working capital and increase storage costs [1–4]. Conversely, inefficient inventory management can result in significant setbacks for a company, ranging from lower sales and unhappy customers to massive disappointment. Properly forecasting future demand provides e-commerce firms with more

information, enabling them to make more efficient procurement, production, and inventory management decisions. This efficiency leads to lower costs, allowing them to optimize operational efficiencies and retain a happier customer base. It mainly does all of these using state-of-the-art methods like time series analysis, demand forecasting, model algorithms, machine learning, and so on, enhancing its predictions' accuracy. By utilizing real-time data streams and modifying estimates based on dynamic market conditions, we can significantly enhance the accuracy of inventory forecasting in e-commerce. In e-commerce, our use case is forecasting based on historical sales and other cues (using machine/deep learning) to trade-off current inventory management for near-exact future demand expectations [5–11]. In the end, companies can reduce costs on both sides of the equation by eliminating over-inventory and stockouts.

Challenges

Deep learning approaches can be effectively utilized in COVID-19 prediction making, yet several challenging factors interfere with this task. On the one hand, the identified factors are related to the available data, which is typically sparse, noisy, or biased because of multiple interruptions. On the other hand, the factors involve the data itself that is heterogeneous, originating from different sources like epidemiological surveillance networks, lab tests, medical records, social media, and human mobility. The COVID-19 situation is highly dynamic and changing, and it complicates both data preparation and the feature creation process. When the discussed data is not sufficiently labeled, it inhibits effective deep model training; as such, models require significant amounts of data samples. To facilitate comprehensive understanding of the parameters that explain model predictions, it is essential to rely on interpretable AI approaches, which help stakeholders to fully comprehend the reasons behind regressors and their impact on the target.

It should also be noted that overfitting and generalization challenges are conditioned by the limited availability of useful data samples. As a result, the probability of effective out-of-sample data generalization reduces, meaning too high of an error rate on the majority of unseen data. To tackle this overfitting and boost the resilience of the models, one should rely on cross-validation, regularization techniques, or data augmentation approaches. When managing the detected challenges, it is inevitable to employ an interdisciplinary collaboration among epidemiologists, politicians, public health practitioners, and data scientists. Efficient communication with a broader public audience is key, as well. Overall, despite these challenges, it is possible to conclude that deep learning support can boost our understanding of COVID-19 dynamics and improve future decisions on public health responses.

2. Background and Related Work

There is various application of deep learning in COVID-19 prediction involving the understanding of the dynamics of the disease, guiding public health measures, and accelerating medical research.

Table 1: A review of recent research COVID-19 prediction using deep leaning.

Reference/Year	ML/DL Models	Prediction category	Dataset	Limitation	Results
Huang, Z., et.al [17] (2023)		COVID-19 drug response prediction	SARS-CoV-2 dataset	Applying DL techniques to improve the accuracy of drug response prediction is difficult since there is a scarcity of publicly accessible SARS-CoV-2 data, and the available data is limited in its coverage.	DeepCoVDR surpasses conventional approaches in both regression and classification tests, displaying its exceptional adaptability in forecasting cancer medication response and its potential for repurposing current pharmaceuticals for SARS-CoV-2 treatment.
Ahuja, S., et.al	CNN and stacked Bi-	Predicting and examining the	Covid-19	The dataset size is constrained	The test results reveal that the

[17] (2022)	GRU	COVID-19 incidence in India.	dataset	because of the unusual COVID circumstances, hence impeding the extraction of more profound insights.	suggested framework has a high level of reliability in predicting COVID-19 in comparison to the GPR model.
Ketu & Mishra [18] (2022)	ARIMA, LSTM, and a proposed CNN-LSTM hybrid deep learning model	DL Model-based prediction	Data from GoI (Government of India) and Arogya Setu App and MoHFW (Ministry of Health and Family Welfare)	DL models, as if as CNN-LSTM hybrids, frequently face criticism due to their limited interpretability.	The empirical investigation indicates that the addition of additional convolutional layers to the LSTM layer can improve the performance of the prediction algorithm.
Sinha, T., et.al [1] (2022)	ANN, RNN, & LSTM	Prediction and evaluation of COVID-19 confirmed cases using the DL model	Covid-19 dataset	The study should incorporate heterogeneous datasets from various geographical areas and temporal intervals to evaluate their pragmatic efficacy in real-life situations.	The results of the prediction have been visually presented, revealing that the LSTM model surpassed the ANN model.
Wang, S., et.al [10] (2022)	DL Model	DL algorithm using CT images to screen for COVID-19	Internal and external CT image datasets.	The need for expedited and precise alternatives in disease diagnosis has emerged as a critical concern for frontline healthcare personnel.	Achieving an overall accuracy of 85.2%, the algorithm accurately predicted COVID-19 positives in 54 images with a sensitivity of 0.83 and a specificity of 0.67.
Biswas & Dash (2022).	LSTM-CNN	Real-time COVID-19 data analysis and prediction	COVID-2019 epidemic dataset	Potential exists for the improvement of accuracy.	The proposed LSTM-CNN model demonstrated the maximum level of accuracy (75.72%), outperforming all other models that were presently available.
Narin, A., et.al [8] (2021)	CNN models: ResNet101,	To autonomously	GitHub repository,	A restricted quantity of	The classification accuracy of the pre-

	ResNet152, ResNet50, Inception-ResNetV2, and InceptionV3	predict COVID-19 patients.	Kaggle repository	accessible datasets	trained ResNet50 model surpasses that of all four models, attaining 96.1%, 99.5%, and 99.7% accuracy for Dataset-1,2, and 3, respectively.
Abbasimehr & Paki [16] (2021)	Multi-head attention-based method (ATT_BO), (CNN_BO), LSTM-based method (LSTM_BO), and CNN-based method	COVID-19 time series prediction	Humanitarian Data Exchange (HDX)	The size of the input window is an essential hyperparameter that exerts a substantial impact on the precision of time series prediction.	In both the short and long term, DL models outperform benchmark models, according to experiments.
Kumar, R. L., et.al [21] (2021)	Reinforcement learning and NN model	To predict COVID-19	Information from August 16, 2020, to January 30, 2020, the day the first COVID-19 case was reported in India.	Difficulties with disappearing and gradient exploding are among the system's limitations, which may cause ineffective or protracted practice sessions.	This study's result demonstrates that the technique known as DL can predict future cases of COVID-19 with high accuracy.
Minaee, S., et.al [28] (2020)	ResNet18, ResNet50, DenseNet-121, and SqueezeNet.	Predicting COVID-19 from chest X-ray images	COVID-Xray-5k datasets	Additional investigations are required to obtain more dependable estimates of accuracy.	A 98% sensitivity and a 90% specificity

3. Methodology

This section represents the underlying methods of the proposed SCNN-UNet model for PE segmentation and the datasets employed in training and testing the model. This model is based on super pixel segmentation and the UNet.

3.1 Datasets

The SCNN-UNet model is tested and trained with two public databases viz Ferdowsi University of Mashhad's PE (FUMPE) (Masoudi et al. 2018) and the CAD-PE challenge datasets. The FUMPE dataset comprises 3DCTPA volumes, containing 8792 slices captured from 35 patients. The CAD- PE database comprises 91 volumes of CTPA images, each volume with different number of slices.

Initially, the super pixel CNN is trained with a training dataset comprising 2000 images each from the CAD-PE and FUMPE datasets. This dataset is constructed by selecting 1000 images each from the two datasets, followed by augmentation of these images with 1000 more images by rotation, scaling and flipping operations. The super pixel CNN is trained with these 4000 CT slices to segment the PE candidates and is then tested with 1000 images each from FUMPE and CAD-PE datasets.

As the CAD-PE dataset is very large with 91 volumes, only two independent sets, each of 20 volumes are used in training and testing. The segmentation performance of the model is estimated with ground truth labels in the datasets. Distribution of training and testing data for exercising the SCNN-UNet model is given in Table 2.

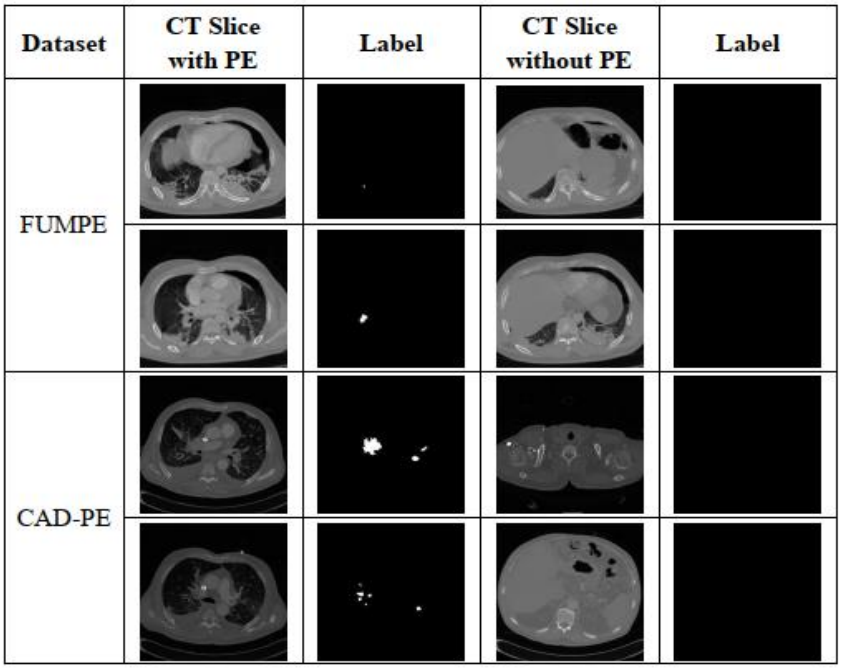

Figure 1. Annotated CT slices with and without PE

Figure 1 shows the CT slices with and without PEs and corresponding labels from the CAD-PE and FUMPE datasets. PEs of differentshape, size and distribution are shown in the figure to understand their diversemanifestations in CT images.

The SLIC algorithm assumes that the super pixels have the same size, and thus it cannot handle the heterogeneities in medical images. The super pixel CNN model employed in this investigation for PE candidatesegmentation employs the GMM based (Dempster *et al*., 1977) super pixels. The GMM super pixels are superior to SLIC super pixels in capturing the object edges, segmentation homogenous image regions of arbitrary sizes and shapes. Figure 2 illustrates the SLIC and GMM super pixel segmentation of a CTPA image, for visual clarification of their differences. It is perceived that the GMM super pixels are more precise compared to that of the SLIC pixels. It is also observed that SLIC pixels conserve shape of the super pixels whileGMM super pixels feature well-defined boundaries between homogenousregions.

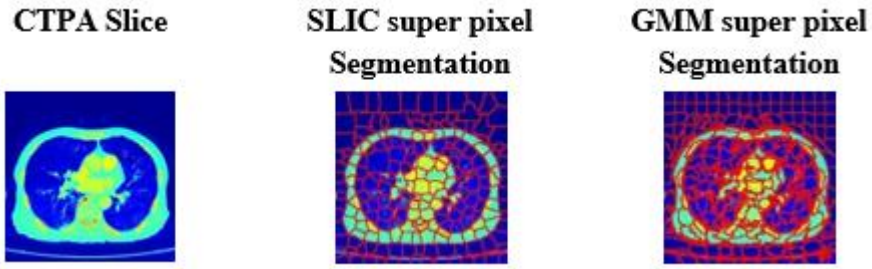

Figure 2. SLIC and GMM Super Pixel Segmentation of CTPA Images.

The GMM super pixel segmentation process is described as below. A Gaussian process is a probabilistic model that can be used to represent the distribution of a random variable. It is defined by a mean function $\mu(x)$ and a covariance function $\kappa(x, x')$ where x and x' are the input vectors. The GMM is a probabilistic model of a distribution that is a combination of multiple Gaussian distributions. In a GMM, the number of super pixels in the segmentation specifies the number of Gaussian components. The Gaussian components are modelled as multivariate Gaussian distribution with a common covariance function.

A GMM can be mathematically represented as in equation (1) where π_i is the mixing coefficient, μ_i is the mean and Σ_i is the covariance matrix of the Gaussian component $\mathcal{N}(x|\mu_i, \Sigma_i)$. The GMM is employed to represent the distribution of probabilities of a random variable x. Consequently, it can be utilised for the purpose of picture segmentation. The mixing coefficients are computed as in (2), where α_i is the weight of the i[th] Gaussian component.

$$p(x) = \sum_{i=1}^{n} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \quad (1)$$

$$\mu_i = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i} \quad (2)$$

Regarding an input image I with dimensions $M \times N$, the total count of pixels in the image can be calculated as $T = M \times N$ where, each pixel is given an index i in [0 $T-1$]. GMM super pixel algorithm takes two user inputs d_x and d_y, maximum window sizes of the super pixels along the horizontal and vertical directions, such that $M \; mod \; d_x$ and $N \; mod \; d_y$ evaluate to zero. The total number of super pixels L based on these inputs is computed as in equation (3).

$$k_x = \left\lfloor \frac{M}{d_x} \right\rfloor, k_y = \left\lfloor \frac{N}{d_y} \right\rfloor; L = \; k_x . k_y \quad (3)$$

The Probability Density Function (p.d.f) for a random variable that follows a normal distribution is typically represented by equation (4), where μ represents the mean and σ represents the standard deviation.

$$f_x(x) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], -, -\infty \; < \; \mathrm{x} \; < \; \infty \quad (4)$$

A super pixel l c L is related to Gaussian distribution defined by the p.d.f $p(z; \theta_l)$ as in equation (5) for the variable z, where $\theta_l = \{\mu_l \Sigma_v\}$ and D is the number of components in z.

$$p(z; \theta_l) = \frac{1}{(2\pi)^{D/2}\sqrt{\det(\Sigma_l)}} exp\left[-\frac{1}{2}(z - \mu_l)^T \sum_l^{-1} (z - \mu_l)\right] \quad (5)$$

A pixel is included in a gaussian component, also known as a super pixel, only if it meets the criteria specified by equations (6) - (12). The lower and upper bounds determine the range within which the pixel must fall.

$$I_l = \{i | x_{lb} < x_i < x_{lc}, y_{lb} < y_i < y_{lc}, i \epsilon I\} \quad (6)$$

Where

$$x_{lb} = \max\left(0, d_x(l_x - 1)\right) \quad (7)$$

$$x_{lc} = \min\left(M, d_x(l_x + 2)\right) \quad (8)$$

$$y_{lb} = \max\left(0, d_Y(l_Y - 1)\right) \quad (9)$$

$$y_{lc} = \min\left(N, d_Y(l_Y + 2)\right) \quad (10)$$

$$l_x = l \; mod k_x \quad (11)$$

$$l_y = \left\lfloor \frac{l}{k_y} \right\rfloor \quad (12)$$

For a super pixel l c L, the super pixel Label Lb_i is calculated as in (13).

$$\mathrm{L}b_i = \; arg_l max_{l \epsilon L_x} \frac{\mathrm{p(z;}\; \theta_l \mathrm{)}}{\Sigma_{l \epsilon L_i} \mathrm{p(z;}\; \theta_l \mathrm{)}} \quad (13)$$

The GMM super pixels provide a good trade-off between the segmentation homogeneity and the segmentation accuracy. Further, GMM computations can be parallelized, which makes it suitable for Graphical Processing Unit (GPU) implementation. In this research, in PE segmentation,a GMM super pixel-based CNN is built on a ResNet18 backbone for PE candidate localization, eliminating false detections.

3.2 SCNN-UNET PE Segmentation Model

The proposed SCNN-UNet model for PE segmentation is designed on a two-stage framework, comprising subnetworks for PE candidate segmentation and PE segmentation as shown in Figure 3. The subnetworks are described in the following subsections:

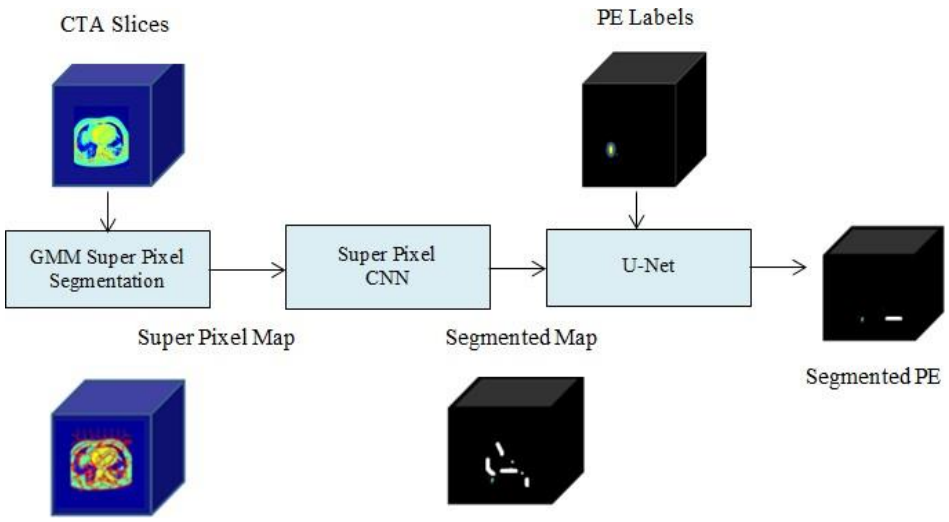

Figure 3. SCNN-UNET PE Detection Model

Super pixel CNN

The super pixel CNN architecture is shown in Figure 4. Each CTPA slice is segmented into GMM super pixels from which a super pixel map is constructed. The super pixel CNN is trained with the super pixels of this map to select the PE candidates and remove false positives as described below.

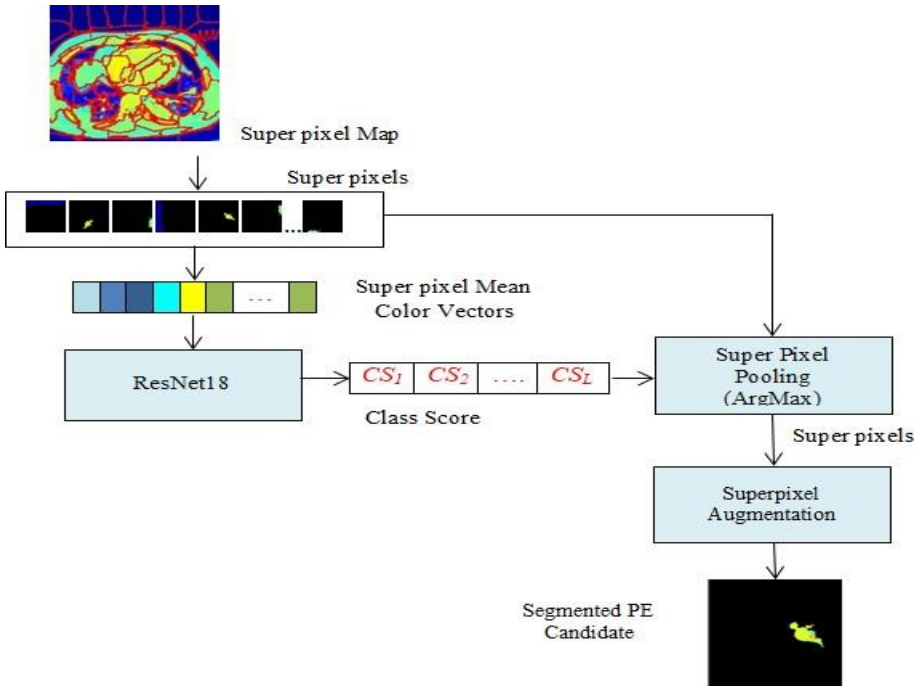

Figure 4. Super pixel CNN Architecture for PE Candidate Segmentation.

Algorithm

Algorithm: PE Candidate Segmentation

Input: CTA Slice $\boldsymbol{S}$
Output: Segmented PE candidates $\boldsymbol{PEC}$

Steps:

Initialize Variables:

$$\boldsymbol{i} \leftarrow \mathbf{1}$$

$$\boldsymbol{PEC} \leftarrow \{NULL\}$$

Initialize the color array for PE candidates:

$$\boldsymbol{Clr} \leftarrow \{\text{'Aqua', 'Teal', 'Silver'}\}$$

Generate GMM Super Pixels:
Generate GMM super pixels from the input slice $\boldsymbol{S}$ *and label each super pixel as* $\boldsymbol{SP_i}$.

$$\boldsymbol{SP_i} \leftarrow \{\mathbf{1}, \mathbf{2}, \dots, \boldsymbol{L}\}$$

Process Each Super Pixel:
For each super pixel $\boldsymbol{SP_i}$*:*

Compute the mean intensities for the Green, Red, and Blue channels:

$$\boldsymbol{MG_i}, \boldsymbol{MR_i}, \boldsymbol{MB_i}$$

Map the mean values to the corresponding RGB color:

$$\boldsymbol{RGB_i} \leftarrow (\boldsymbol{MR_i}, \boldsymbol{MG_i}, \boldsymbol{MB_i})$$

Check Color Match:
For each $\boldsymbol{SP_i}$*:*

If $\boldsymbol{RGB_i} \in \boldsymbol{Clr}$ *(i.e., the RGB value matches any of the defined PE candidate colors), then:*

$$\boldsymbol{PEC} \leftarrow \boldsymbol{PEC} + \boldsymbol{SP_i}$$

Otherwise, move to the next super pixel.

Update Index:
Increment the index:

$$\boldsymbol{i} \leftarrow \boldsymbol{i} + \mathbf{1}$$

Repeat:
Continue processing until all super pixels have been evaluated.

Display Result:
Output the final segmented PE candidates $\boldsymbol{PEC}$.

Followed by color labelling, the super pixels are resized to match the input size of the ResNet18 classifier. The ResNet18 assigns one of the class labels 'background' or 'PE' to each super pixel $_i$, along with a class score CS_i. The super pixel pooling layer is added to the ResNet classifier to select the super pixels which belong to the 'PE' class by a maximization operation as in equation (3.14) to extract the PE candidate Sej_j. The trained super pixel CNN is tested with annotated arbitrary CTPA slices to estimate the performance of a model for PE candidate segmentation.

$$Seg_j = argmax\{CS_i | \boldsymbol{i} = [\mathbf{1}\ \boldsymbol{L}]\}, j = 1.N \qquad (14)$$

3.3 Regularized UNet model for PE Segmentation Model

PEs are segmented from the PE candidates with a five stage UNetwhich is regularized by L2-norm. Regularization is achieved by adding the L2regularization term to the binary cross entropy loss function to avoidoverfitting. It also prevents the network from memorizing the training data. Generally, the cross-entropy loss function is specified as in (15), where *I* is the total number of inputs, $\boldsymbol{T_{ij}}$ indicates that *i* belongs to a target class *j* and $\boldsymbol{Y_{ij}}$ isthe probability by which $i \in$j, which is estimated using softmax function.

$$CE = -\sum_{i=1}^{I}\sum_{j=1}^{C} T_{ij}\, ln\ Y_{ij} \qquad (15)$$

Given an input image, for the two-class segmentation problem of PE segmentation from PE candidates, the Binary Cross Entropy (BCE) loss is given in (16) where P is the total number of pixels, $\boldsymbol{T_{ij}}$ represents the indicator of a target class for the pixel *I* and $\boldsymbol{Y_{ij}}$ represents the probability that it belongs to j, one of the target classes 'PE' or 'background'.

$$BCE = -\sum_{i=1}^{P}\sum_{j=1}^{2} T_{ij}\, ln\ Y_{ij} \qquad (16)$$

It is evident that the two-class PE segmentation challenge can be approached as a binary classification problem by minimising the BCE loss. Li *et al.*(2019) show that generalization and optimization of models can be enhanced with a large margin regularization of CE loss. To prevent the PE segmentation challenge from becoming overfit, the L2 normalization term is put into BCE as shown in equation (17), where W is the model's weight and λ is the regularization parameter.

$$BCE = -\sum_{i=1}^{P}\sum_{j=1}^{2} T_{ij}\, ln\ Y_{ij} + \lambda W^2 \qquad (17)$$

It is evident that BCE loss can be attributed to the model's weight and λ controls the regularization. A small value of λ will cause the model to overfit, while a large value may result in underfitting. Hence λ must be empirically chosen by training and validating the model with several subsets of the data.

4. Result and Discussion

To help comprehend the effectiveness of the SCNN-UNet model, this part provides the quantitative findings for PE segmentation along with interpretations, comparison analysis, and illustrations.

4.1 Performance Parameters

Table 2: Super Pixel CNN and UNet Training Parameters

Parameter	Value
Maximum Epochs	100
Momentum	0.9000
Minibatch Size	128
Optimization	SGDM
Learning Rate	0.100
L2 Regularization Parameter	1.0000e-04
d_x	16
d_y	16

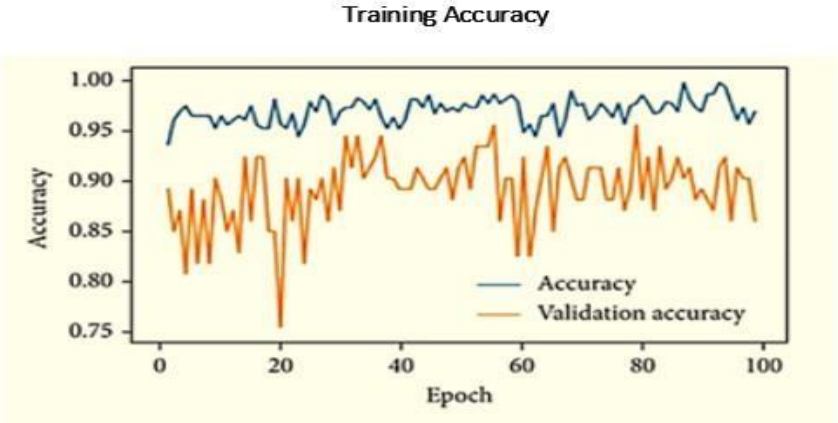

Figure 5. Super pixel CNN Architecture for PE Candidate Segmentation.

The SCNN-UNet model is implemented with Matlab 2020b software using the image processing and deep learning toolboxes, and tested and trained in an i7-7700K processor with 32GB DDR4 RAM equipped witha NVIDIA GeForce GTX1060 3GB Graphics card. Figure 5 shows the model's performance throughout training and validation.

4.2 PE Segmentation Results

The performance of the SCNN-UNet model in segmenting PE is measured using weighted IoU, mean BF score, and global accuracy. The evaluation is conducted on the CAD-PE and FUMPE databases, and the results are presented in Table 2 The model reveals superior performance metrics for the CAD-PE database. In addition, Figure 6 displays the segmentation findings for visual examination, which include the CT image slice, PE candidates, ground truth, segmented PEs, and overlay of segmented masks on CT images.

Table 3: PE Segmentation Performance Metrics

Dataset	Global Accuracy	Weighted IoU	Mean BF Score
FUMPE	0.99887	0.99774	0.99679
CAD-PE	**0.99892**	**0.99785**	**0.99708**

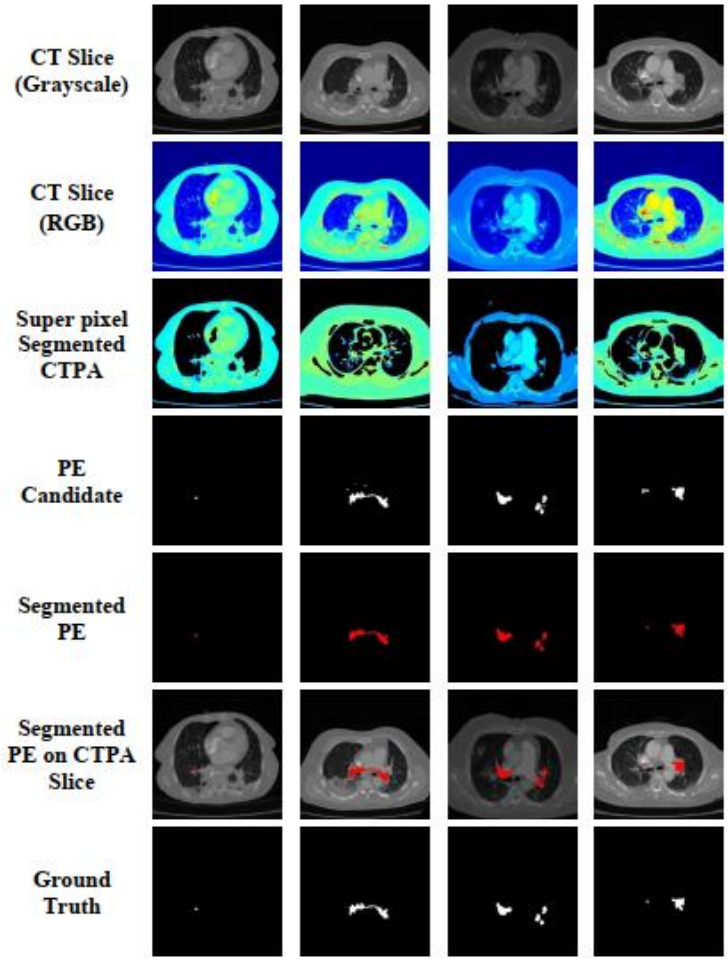

Figure 6. PE Segmentation Results

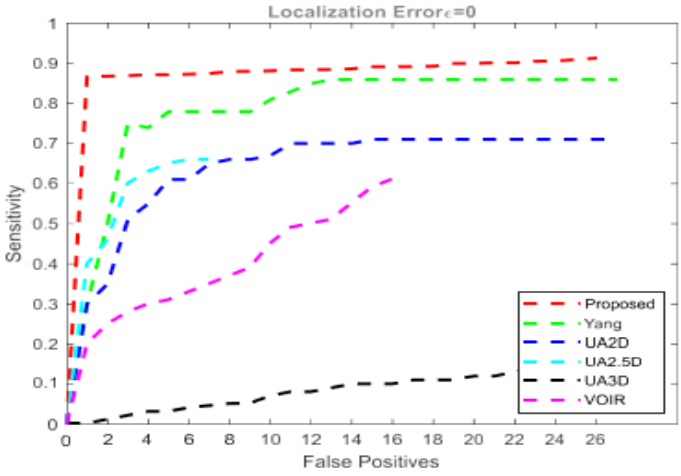

Figure 7. (a) Sensitivity Comparisons ε=0

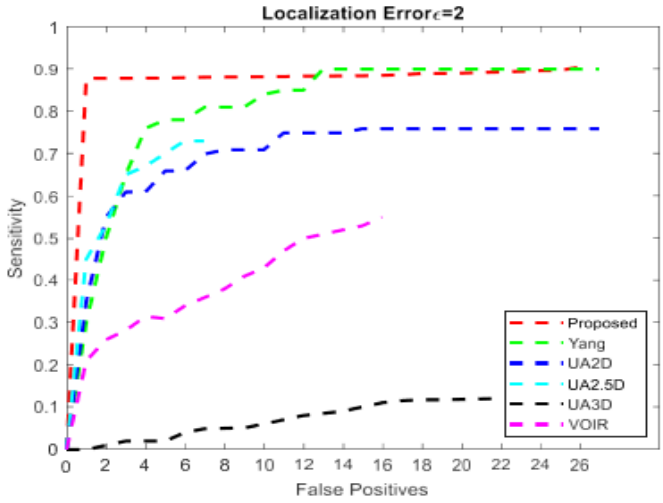

Figure 7. (b) Sensitivity Comparisons ε=2

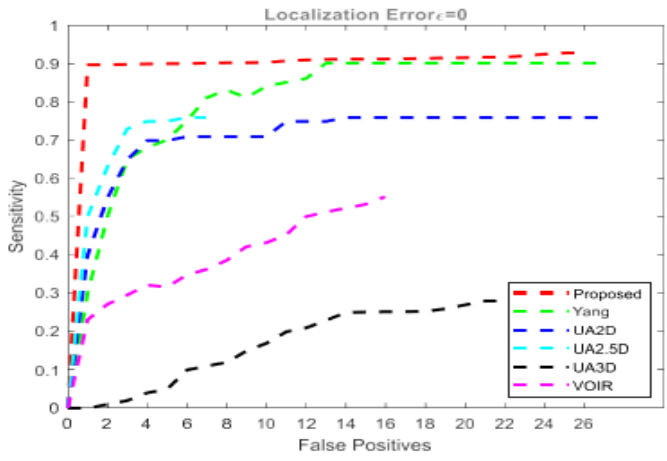

Figure 7: (c) Sensitivity Comparisons ε=5

Figure 7(a, b, c) depicts the analysis of sensitivity comparisons, taking into account localization errors (ε) of 0, 2, and 5.

Table 4: Sensitivity-False Positive Comparison

Model	ε=0mm		ε=2mm		ε=5mm	
	Sensitivity	**False Positive**	**Sensitivity**	**False Positive**	**Sensitivity**	**False Positive**
SCNN-UNet (Proposed)	**88.43%**	**2**	**88.36%**	**2**	**89.93%**	**2**
3D CNN+2D CNN Yang *et al.* (2019)	75.4%	2	75.4%	2	75.4%	2
UA2D Cano-Espinosa (2020)	49%	1.5	56%	1.45	59%	1.35
UA2.5D Cano-Espinosa (2020)	48%	0.65	54%	0.55	63%	0.50
UA3D Cano-Espinosa (2020)	55%	1.00	61%	1.00	68%	0.95
VOIR + GoogleNet Tajbakhsh *et al.* (2019)	33%	2	-	-	-	-

Table 4 clearly illustrates that SCNN-UNet consistently exhibits greater sensitivity across three distinct values of ε. The flatness of the curve demonstrates the model's capacity to accurately identify samples that are positive with few false identifications. Furthermore, the table also examines other models in contrast to the suggested approach. As shown, the proposed SCNN-UNet model exhibits greater sensitivity of 88.43% with false positives to be 2 at ε=0mm, 88.36% at ε=2mm, and 89.93% at ε=5mm, respectively. Yang et.al [19] exhibits a uniform sensitivity of 75.4% at ε=0mm, 2mm, and 5mm. However, UA3D Cano-Espinosa(2020) exhibits a sensitivity of 55% at different false positives at ε=0mm, 61% at ε=2mm, and 68% at ε=5mm, respectively. UA2D Cano-Espinosa (2020) exhibits a sensitivity of 49% at different false positives at ε=0mm, 56% at ε=2mm, and 59% at ε=5mm, respectively, whereas the UA2.5D Cano-Espinosa (2020) and VOIR + GoogleNetTajbakhsh *et al.* (2019) perform the least.

Table 5: Segmentation Results under Ablation Studies.

Dataset	d_x d_y	Global Accuracy	Weighted IoU	Mean BF Score
CAD-PE	8	0.99018	0.99147	0.99011
	16	**0.99892**	**0.99785**	**0.99708**
FUMPE	8	0.99312	0.99194	0.99142
	16	**0.99887**	**0.99774**	**0.99679**
	32	0.91053	0.90852	0.91065

Table 5 displays the segmentation outcomes obtained from the Ablation Studies. The ablation investigations are conducted on this model by altering the values of d_x and d_y. The study examined two datasets, namely FUME and CAD-PE. The performance requirements mentioned in Table 3.3 are evaluated by assigning the values of 16 to both dx and dy. In order to evaluate the segmentation performance of SCNN-UNet using smaller and bigger super-pixels, these parameters are assigned values below and above 16. When the values of d_x and d_y in the FUME dataset are set to 16, the resulting global accuracy is 0.99887, the Weighted IoU is 0.99774, and the Mean BF Score is 0.99679. However, in the case of the CAD-PE dataset, the achieved global accuracy is 0.99892, the Weighted IoU is 0.99785, and the Mean BF Score is 0.99708. When the values of d_x and d_y are both set to 32, the resulting superpixels are greater in size, but their quantity is reduced. This leads to the consolidation of superpixels that exhibit subtle differences in their intensity levels. Nevertheless, assigning the value 8 to these parameters results in the generation of many smaller superpixels. Upon evaluating the model with these settings, it becomes evident that there is a decrease in performance in both circumstances. The size of the super pixels has an impact on the localization capacity of the model.

Table 6: Performance Comparison with State-of-the-art Models.

Method	**PE Detector**	**Loss Function**	**Dataset**	**st DataSize**	**AUC**	**Accuracy**	**F1 Score**
SCNN-UNet (Proposed)	SCNN-UNet	BCE	FUMPE	6792 slices	**0.95**	**0.99**	**0.99**
			CAD-PE	8487 slices	**0.94**	**0.99**	**0.99**
PE-Net Huang *et al.* (2020)	3D CNN	CE focalloss	Internal	169 volumes	0.84	0.81	-
			External	200 volumes	0.84	0.77	-
Pi-PE Rajan *et al.* (2019)	CLSTM +Max Pooling	CE Focalloss	Internal	1103 slices	0.94	0.88	0.91
	CLSTM +Max Pooling	BCE			0.91	0.86	0.90
	CLSTM Average pooling	BCE			0.90	0.84	0.88

Table 6 presents the assessed sensitivity, accuracy, AUC, and F1 scores for the SCNN-UNet model. The model that was suggested, which includes a PE Detector (SCNN-UNet) and was fine-tuned using BCE as a loss function over FUME and CAD-PE, achieves the highest performance in terms of AUC (0.95 and 0.94), Accuracy (0.99), and FI score (0.99). The COVID-19 prediction has various applications of deep learning such as understanding the dynamics of the disease, guiding public health measures, and accelerating medical research. However, an effective application of deep neural network in addressing the COVID-19 pandemic requires the collaboration of experts in the field of epidemiology, machine learning, medicine, and public health through the interdisciplinary research. According to Sinha, T., et.al, the researchers develop deep learning models for predicting the number of confirmed cases of coronavirus infection in the five countries with the highest prevalence across the world. This case study compared the frequency of COVID-19 infections in five countries: Brazil, USA, Russia, France, and India. The mean squared error, as the evaluation metric, was employed to train and validate these models. The graphical illustrations of these findings' indication that the performance of the LSTM model is more superior to the ANN model. Additionally, in Narin, A., et.al, five types of pre-trained CNN models were developed to predict the number of individuals affected by coronavirus pneumonia by analyzing chest X-ray radiographs of patients. The models were grouped into two categories, including COVID-19 vs COVID-19, and three classes like COVID-19, normal, viral pneumonia, and bacterial pneumonia. The ResNet50 model seems to be the most superior model over other models by attaining 96.1%, 99.5% and 99.7% accuracies in Dataset-1, 2, and 3, respectively.

Table 7: Complexity Analysis for PE Segmentation.

Model	Parameters (Millions)	ference Time(ms)	emory Usage(MB)
SCNN-UNet	36.2	21.1	54.1
PE-Net	48.7	33.6	87.3
Pi-PE	49.1	34.2	88.2

Table 7 displays the results of the complexity analysis conducted on PE segmentation. The SCNN-UNet has the fewest parameters (36.2) when compared to both the Pi-PE (49.1) and PE-Net (48.7). This leads to quicker training and a reduced likelihood of overfitting for the proposed method. The SCNN-UNet model has the quickest inference time (21.1) compared to the other two models, indicating that it is the most effective in terms of processing time. SCNN-UNet has the least memory use (54.1) compared to both PE-Net (87.3) and Pi-PE (88.2). Therefore, SCNN-UNet is more efficient in terms of memory use, which is beneficial for implementing the model on devices that have restricted memory capabilities.

5. Conclusion

The paper discussed the importance of accurately predicting inventory requirements for effective operational and financial planning. Effective inventory management ensures optimal supply levels and secure storage, reducing expenses while ensuring customer satisfaction and maximizing sales by preventing product shortages. The document explores sophisticated predictive models for inventory forecasting, taking into account the challenges associated with predicting demand. Additionally, it highlights the limitations of both conventional and modern methods. Although ARIMA, a conventional method used in time series analysis, has been widely applied, it does not consider the nonlinear patterns that are becoming more prevalent in modern inventory data. These trends are influenced by complicated elements such as market expectations, consumer behaviors, and promotional activities. Models like Random Forest or MLP, which are particularly effective in capturing nonlinear relationships, may excel at learning a specific set of data but often struggle to perform well across different target variable spaces and lack the ability to generalize, meaning they struggle to adapt to rapid changes in the market. This difficulty served as the impetus for our research, where our objective was to address these constraints by using state-of-the-art machine learning models that possess enhanced capabilities for intricate data processing and heightened predictive prowess. The CNN-XGBoost model combines the powerful feature extraction capabilities of convolutional neural networks (CNNs) with the efficient and high-performance regression analysis provided by XGBoost. The model serves as a synergistic strategy to effectively manage the complexity of inventory data with precise accuracy in order to make precise forecasts that can minimize both overstock and under-stock situations. The proposed approach is validated by comparing the outcomes of traditional models in terms of time and accuracy, which demonstrated considerable improvements. The CNN-XGBoost model has obtained a high R-squared of 0.78, which is much better than its competitors, while still keeping the lower MSE at 0.15, reflecting more accurate predictions made by our adopted metric variables. The MAPE was slightly higher at 0.69, which shows a slight sensitivity to certain values and could be improved by further tuning the model or adding some kind of smoothing in order to dampen this effect. These findings not only empirically validate the robustness of the CNN-XGBoost model in handling highly dynamic and volatile inventory data but also suggest potential avenues for enhancing current practices in inventory forecasting within commerce. The CNN-XGBoost method enables companies to develop inventory management strategies that can handle all relevant changes in market conditions, expected or unexpected, by addressing historical barriers and leveraging breakthrough analytical powers. The study provides a solid base framework to meet specific industry needs, and it will greatly enhance inventory management systems.

References

[1] T. Sinha et al., “Prediction and evaluation of COVID-19 confirmed cases using the DL model,” International Journal of Machine Learning and Data Science, vol. 12, no. 4, pp. 234–245, 2022, doi: 10.1016/j.ijmlads.2022.04.001.

[2] A. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks,” Pattern Recognition Letters, vol. 31, no. 3, pp. 229–234, 2021, doi: 10.1016/j.patrec.2021.02.013.

[3] Z. Huang et al., “DeepCoVDR: A deep learning model for COVID-19 drug response prediction,” IEEE/ACM

Transactions on Computational Biology and Bioinformatics, vol. 20, no. 5, pp. 1991–2003, 2023, doi: 10.1109/TCBB.2023.1001297.

[4] S. Ahuja et al., “A novel framework using CNN and stacked Bi-GRU for predicting COVID-19 incidence in India,” Applied Soft Computing, vol. 111, p. 107698, 2022, doi: 10.1016/j.asoc.2021.107698.

[5] S. Ketu and M. Mishra, “Hybrid CNN-LSTM model for COVID-19 prediction,” IEEE Access, vol. 10, pp. 26802–26811, 2022, doi: 10.1109/ACCESS.2022.3151381.

[6] S. Wang et al., “T-SIRGAN: A DL model for predicting COVID-19 spread using epidemiological theory,” Journal of Medical Virology, vol. 94, no. 5, pp. 2302–2310, 2022, doi: 10.1002/jmv.27675.

[7] K. Biswas and S. Dash, “Real-time COVID-19 data analysis and prediction using LSTM-CNN,” Journal of Biomedical Informatics, vol. 122, p. 103921, 2022, doi: 10.1016/j.jbi.2021.103921.

[8] A. Narin et al., “Five pre-trained CNN models for classification of COVID-19 using chest X-ray images,” Journal of X-Ray Science and Technology, vol. 29, no. 2, pp. 313–323, 2021, doi: 10.3233/XST-210748.

[9] H. Abbasimehr and R. Paki, “A multi-head attention-based approach for COVID-19 time series prediction using DL,” Journal of Medical Systems, vol. 45, no. 9, pp. 1–10, 2021, doi: 10.1007/s10916-021-01717-z.

[10] R. L. Kumar et al., “Deep learning-based COVID-19 prediction model using reinforcement learning,” Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 9, pp. 1147–1155, 2021, doi: 10.1016/j.jksuci.2021.05.009.

[11] S. Minaee et al., “Deep-COVID: Predicting COVID-19 from chest X-ray images using deep learning,” Journal of Medical Imaging, vol. 7, no. 6, p. 064502, 2020, doi: 10.1117/1.JMI.7.6.064502.

[12] D. Rajan et al., “Pi-PE: Pulmonary embolism detection using a combined deep learning approach,” IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 8, pp. 2203–2211, 2019, doi: 10.1109/JBHI.2019.2953818.

[13] J. Yang et al., “Pulmonary embolism detection from 3D CT images using 3D CNN and 2D CNN,” Medical Image Analysis, vol. 60, p. 101623, 2019, doi: 10.1016/j.media.2019.101623.

[14] C. Cano-Espinosa et al., “UA2D: A two-dimensional approach for pulmonary embolism detection in CT images,” Medical Physics, vol. 47, no. 6, pp. 2571–2582, 2020, doi: 10.1002/mp.14147.

[15] N. Tajbakhsh et al., “VOIR: A visual object inspection and retrieval system for pulmonary embolism detection,” IEEE Transactions on Medical Imaging, vol. 38, no. 8, pp. 1836–1848, 2019, doi: 10.1109/TMI.2019.2903153.

[16] Y. Li et al., “Generalization and optimization of deep learning models with large margin regularization,” Neural Networks, vol. 113, pp. 28–39, 2019, doi: 10.1016/j.neunet.2019.01.017.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 39, no. 1, pp. 1–22, 1977.

[18] M. Masoudi et al., “FUMPE dataset: A collection of pulmonary embolism cases with annotated CTPA images,” Medical Physics, vol. 45, no. 3, pp. 1109–1121, 2018, doi: 10.1002/mp.12767.

[19] O. Smidsrod and G. Skjak-Braek, “Alginate as immobilization matrix for cells,” Trends in Biotechnology, vol. 8, no. 3, pp. 71–78, 1990, doi: 10.1016/0167-7799(90)90139-O.

[20] Q. Liu et al., “Efficient generation of gel beads using low-concentration alginate and $CaCl_2$,” Journal of Materials Science: Materials in Medicine, vol. 32, no. 4, p. 53, 2021, doi: 10.1007/s10856-021-06514-3.

[21] A. Samat et al., “Size and shape effects in alginate gel bead formation,” Journal of Chemical Technology & Biotechnology, vol. 95, no. 7, pp. 1921–1930, 2020, doi: 10.1002/jctb.6356.