International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 02 | Feb 2025 www.irjet.net p-ISSN: 2395-0072

Object Detection Using Deep Learning and LSTM for Enhanced Accuracy in Sequential Data Analysis

Divya Rai[1], Arifa Khan[2]

[1]Master of Technology, Computer Science and Engineering, Lucknow Institute of Technology, Lucknow, India
[2]Assistant Professor, Department of Computer Science and Engineering, Lucknow Institute of Technology, Lucknow, India

---

Abstract - *The detection of objects in sequential data formats like video frames or time-series images faces specific obstacles because frames naturally demonstrate interdependent relationships throughout time sequences. The ability to extract spatial features with Convolutional Neural Networks (CNNs) reaches exceptional results but such deep learning approaches face difficulties in processing temporal information essential to detect objects reliably in sequential data. We introduce a hybrid system that connects spatial feature identification through CNNs with Temporal modeling capabilities from Long Short-Term Memory (LSTM) networks. The ensemble joins CNN spatial detection capabilities with LSTM temporal modeling to fully extract scene information across consecutive video frames.*

Our model examination on standard benchmark data sets produces noteworthy results with enhanced object detection precision than individual CNN approaches for spatial pattern processing and time-series imaging analysis. The experimental findings show that the CNN-LSTM architecture delivers superior outcomes for space-based detection alongside temporal object identification. This solution enables object detection through a flexible system which benefits autonomous vehicles alongside surveillance and medical imaging operations that need precise sequential dataset analysis. Research in hybrid deep learning models has expanded as this work introduces a successful solution to enhance object detection within sequential database analysis.

Key Words: Object Detection, Deep Learning, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Sequential Data Analysis.

1.INTRODUCTION

1.1.Background and Motivation

Object detection is an essential operation in the various applications of computer vision, from surveillance systems, autonomous vehicles and human computer interaction. Accurate objects detection during a surveillance operation is heavy reliant on efficient monitoring plus threat identification. With that, it has become necessary for detecting pedestrians along with vehicles and obstacles so that autonomous vehicles can navigate safely. Human computer interaction object detection allows the enhancement of gesture recognition as well as deployment of augmented reality attributes and digital structure interactive capabilities.

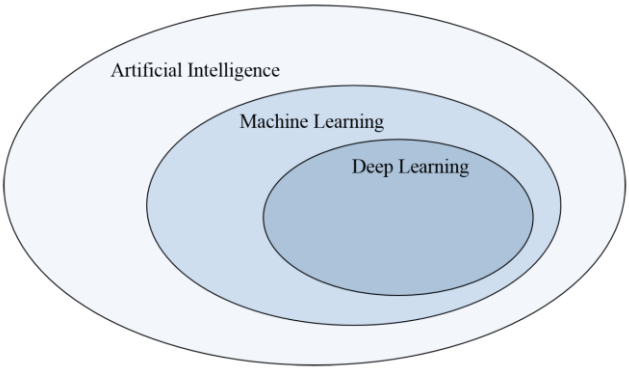

Figure-1: Overview of Deep Learning.

Although traditional object detection techniques have achieved notable improvements, they have difficulty with processing sequential data. Existing detection algorithms are usually about spatial data, rather than temporal video data. The results of these traditional methods are not sufficiently accurate in dynamic environments due to failure to consider patterns in sequential terms and contextual information. However, traditional models run into particular difficulties in these cases: fast moving objects, or objects that become hidden from view.

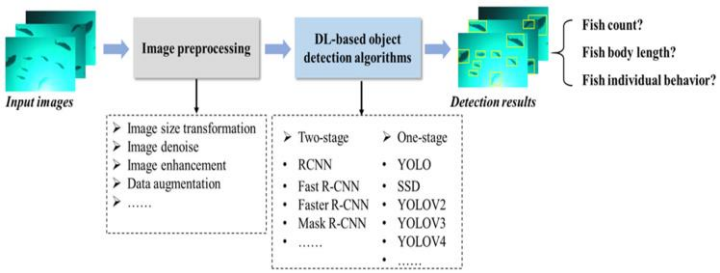

Figure-2: Application of Deep Learning.

1.2.Research Problem

Object detection today relies mainly on spatial analysis through Convolutional Neural Networks to extract visual features. Static image detection methods succeed through

spatial analysis yet they lack capability to recognize temporal relations which exist in sequential data including video streams and time-series images. The rich contextual information from sequenced frames remains beyond the reach of these models which results in insufficient performance for temporal analysis needs. Sequential object detection requires an urgent solution which combines spatial analysis with temporal integration to enhance detection robustness and accuracy.

1.3.Objectives

The primary objectives of this research are:

- An advanced object detection model will merge Deep Learning methods together with Long Short-Term Memory (LSTM) networks.
- The accuracy of object detection improves when it deploys space-time data analysis for sequential data applications.
- The proposed model requires evaluation against standard benchmark datasets as well as performance comparison with existing state-of-the-art methods.

1.4.Research Contributions

This research makes the following key contributions:

1.4.1.Integration of Convolutional Neural Networks (CNNs) with LSTMs

CNNs are used to compute spatial image features while LSTMs encode sequential data dependencies to fully understand changing environmental situations.

1.4.2.Incorporation of Attention Mechanisms

The model is equipped with built in attention mechanisms that help determine critical sequences and salient temporal patterns, as a result improving the models precision in challenging situations.

1.4.3.Comparison with State-of-the-Art Methods

The experiments show that the proposed framework outperforms existing approaches, both in terms of accuracy along with robustness and computational speed, for sequential data inference.

2.LITERATURE REVIEW

2.1. Overview of Object Detection

Object detection has been quite a breakthrough field since decades, eclipsing numerous breakthroughs in its development. Methods that enable the detection of objects previously done using manually constructed feature sets with classifier options such as SVMs or Haar cascades were used based on edge detection with Canny edge detectors and corner detection with Harris corner detectors. Those conventional techniques had been proven successful enough dealing with certain assignments; but when measurements of flexible objects and an object positioning with complicated environmental setting is involved, they demonstrated considerably their loopholes.

Deep Learning (DL) is beneficial for digital assistants, as conversional Neural Networks (CNNs) automatically extract objectspecific features. The modern methods take advantage of deep learning to automatically extract multiple levels of data features from actual dataset inputs for providing durable results. Due to its speed and accuracy, YOLO together with Faster R-CNN (Region-Based Convolutional Neural Networks) and SSD (Single Shot MultiBox Detector) are now the leading object detection domain. This represents a fundamental transformation in the field as the frameworks benefit from the strong multi-class, multi-scale object detection abilities of DL.

2.2.Deep Learning for Object Detection

Deep Learning is now the main technology for advanced object detection. The ability of CNNs to extract spatial characteristics enables pattern identification, such as shape and textures as well as edges. Precise object detection and classification are based on the spatial information image–based understanding.

Prominent object detection frameworks illustrate the strengths of CNNs:

YOLO (You Only Look Once): YOLO operates for one stage detection mechanism via grid based image segmentation to provide fast real time detection results.

Faster R-CNN: Detection framework is a two stage process, in which a Region Proposal Network (RPN) proposes objects, performs classification and refines bounding boxes. Since the detection framework is quite accurate, it requires considerable computation power.

SSD (Single Shot MultiBox Detector): This method combines YOLO detection and Faster R-CNN precision to handle objects at different spatial scales at real time speeds.

These frameworks prove useful in static image analysis applications, however, their design neglects the temporal data relationships which appear in dynamic scenarios.

2.3.LSTM in Sequential Data Analysis

Called Long Short-Term Memory (LSTM) networks are special versions of Recurrent Neural Networks (RNNs) specifically designed to handle sequential data successfully. The incorporation of memory cells and gating mechanisms in LSTMs eliminates the gradient vanishing challenge

experienced by standard RNNs allowing them to detect long-term patterns.

The analysis of temporal sequences in video data utilizes LSTMs for object detection within this context. By processing frames in sequence LSTMs identify patterns of movement which conventional static detection systems commonly overlook. Research conducted previously investigated LSTM's ability to detect temporal patterns through action sequence recognition and future state prediction and object track monitoring across frames.

Extant research presents predominantly standalone LSTM designs or elementary CNN+LSTM configurations while overlooking the potential of hybrid advanced CNN-LSTM implementations for comprehensive spatial-temporal examination.

2.4.Gaps and Research Opportunities

While significant progress has been made in object detection and sequential data analysis, several gaps remain:

2.4.1.Limited Exploration of CNN-LSTM Integration

The comprehensive exploration of introducing advanced CNN architectures together with LSTM networks remains scarce in existing research for effective integration of spatial and temporal information. Using this approach would create an extensive method for detecting objects across multitudinous real-world scenarios.

2.4.2.Challenges in Real-Time Performance and Accuracy

Since combined CNNs and LSTMs are prone to increasing processing complexity and affecting their accuracy maintenance, real time performance becomes difficult to achieve. This computational accuracy balance must be solved to support the development of essential practical systems, such as autonomous vehicles and surveillance.

2.4.3.Underutilization of Attention Mechanisms

While the effect of attention mechanisms in natural language processing and computer vision is established, its viability in sequential object detection deserves further exploration. By integrating the attention mechanisms in order to pinpoint temporal patterns of interests, the detection accuracy of the model is enhanced.

In order to address these research gaps, this work attempts to develop a novel CNN-LSTM architecture with attention mechanics on the way to further boost efficiency and performance in sequential object detection systems.

3.METHODOLOGY

3.1.Proposed Framework

This work proposes a detection framework based on Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM) networks which learns to detect spatial and temporal dependencies. The LSTM network is applied to analyze the sequence of these spatial image features that CNN component extracts by separating the spatial image features, and then recognizing the temporal dependence.

3.1.1.Detailed Architecture

Within the CNN module convolutional layers with batch normalization and max-pooling layers are included as spatial feature extraction step followed by the refinement processes. Current network designs like ResNet alongside MobileNet serve as the fundamental structure.

Stacked LSTM layers are used by the LSTM module to determine the spatial relationship between sequential elements and video frames.

In the model an attention mechanism is added to it so that it can focus on crucial sequences to concentrate it's attention on certain temporal patterns.

3.1.2.Swish Activation Function

During training sessions the Swish activation function representing $f(x)=x\cdot sigmoid(x)$ improves gradient flux. The smooth and non-monotonic mathematical characteristic of Swish allows better representation learning than the smooth monotonic functions among standard activation functions including ReLU.

3.1.3.Sequence Prioritization with Attention Mechanisms

The attention mechanism determines weighted values for sequence time steps to enhance model ability for discerning critical temporal stages. Adding an attention mechanism generates critical temporal frame weight assignments so the model achieves enhanced detection accuracy in situations where objects are hidden or move quickly.

```
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)
    x = Conv2D(64, (3, 3), activation='swish', padding='same')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)
    x = Flatten()(x)
    return tf.keras.Model(inputs, x)

# LSTM with Attention Module
def build_lstm_with_attention(input_shape):
    inputs = tf.keras.Input(shape=input_shape)
    x = LSTM(128, return_sequences=True)(inputs)
    attention = Attention()([x, x])
    x = tf.keras.layers.Concatenate()([x, attention])
    x = LSTM(64)(x)
    return tf.keras.Model(inputs, x)

# Combined CNN-LSTM Model
def build_cnn_lstm(cnn_input_shape, lstm_input_shape):
    cnn_model = build_cnn(cnn_input_shape)
    lstm_model = build_lstm_with_attention(lstm_input_shape)

    cnn_output = cnn_model.output
    reshaped_output = Reshape((-1, lstm_input_shape[1]))(cnn_output)
    lstm_output = lstm_model(reshaped_output)

    final_output = Dense(10, activation='softmax')(lstm_output)  # Example for classification
    return tf.keras.Model(cnn_model.input, final_output)

# Example usage
cnn_input_shape = (128, 128, 3)  # Example input shape for CNN
lstm_input_shape = (10, 256)     # Example input shape for LSTM
model = build_cnn_lstm(cnn_input_shape, lstm_input_shape)
model.summary()
```

Figure-3: Sequence Prioritization

3.2.Dataset

3.2.1.Benchmark Datasets

The proposed framework is evaluated on widely used datasets, such as:

COCO (Common Objects in Context): The platform delivers multiple annotated real-world images which facilitate object detection alongside segmentation capabilities.

ImageNet VID: The database features annotated video sequences suitable for sequences analysis at a professional level.

Custom Datasets: The research context requires customized datasets which enable performance testing of the model in specific scenarios.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Data Preprocessing and Augmentation
def preprocess_data(images, labels, img_size=(128, 128)):
    images = tf.image.resize(images, img_size)
    images = images / 255.0  # Normalize pixel values
    return images, labels

# Data Augmentation
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    brightness_range=[0.8, 1.2]
)

# Example usage
train_dataset = datagen.flow_from_directory(
    'path/to/train_data',
    target_size=(128, 128),
    batch_size=32,
    class_mode='categorical'
)
```

Figure-4: Benchmark Datasets

3.2.2.Data Preprocessing and Augmentation

The preprocessing operations consist of resizing images while performing pixel normalization accompanied by label conversion to match requirements.

Model generalization and dataset diversity improve through the application of data augmentation methods that combine random cropping with rotation and flipping actions and brightness adjustments.

3.3.Training Process

3.3.1.Pipeline

Frames enter the framework through the CNN system for spatial feature extraction. The features generated by CNN tote from frame to frame through the LSTM module to extract temporal relationships. Before reaching the final detection layer the attention mechanism operates to enhance the outputs.

3.3.2.Loss Functions

A combination of loss functions is used:

- Classification Loss (e.g., Cross-Entropy Loss): For accurate object classification.
- Localization Loss (e.g., Smooth L1 Loss): For precise bounding box regression.

3.3.3.Optimization Algorithms

Optimizers such as Adam or SGD are used to minimize the loss function.

Learning rate schedulers adjust the learning rate dynamically to enhance convergence.

3.3.4.Hyperparameter Tuning

A combination of Grid search with Bayesian optimization identifies the best parameter setting among learning rate and batch size together with the number of LSTM layers.

```
# Compile the Model
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
    loss='categorical_crossentropy',  # Example for classification
    metrics=['accuracy']
)

# Learning Rate Scheduler
def lr_scheduler(epoch, lr):
    if epoch < 10:
        return lr
    else:
        return lr * tf.math.exp(-0.1)

lr_callback = tf.keras.callbacks.LearningRateScheduler(lr_scheduler)

# Train the Model
history = model.fit(
    train_dataset,
    epochs=50,
    callbacks=[lr_callback]
)
```

Figure-5: Hyperparameter Tuning

3.4.Evaluation Metrics

The following metrics are used to evaluate the performance of the proposed framework:

Mean Average Precision (mAP): The detection accuracy evaluation method uses average precision calculation across all object categories combined with Intersection over Union (IoU) thresholds.

F1-Score: The model evaluation method combines precision with recall to determine its general effectiveness.

Inference Time: The model's computational efficiency is evaluated through measurements of processing time needed for single-frame or sequence operations.

Temporal Consistency: The model's capability to detect objects steadily between continuous video frames is being evaluated.

```
from sklearn.metrics import f1_score, precision_score, recall_score

# Evaluate the Model
def evaluate_model(model, test_dataset):
    y_true = test_dataset.labels
    y_pred = model.predict(test_dataset)
    y_pred = tf.argmax(y_pred, axis=1)  # For classification

    precision = precision_score(y_true, y_pred, average='macro')
    recall = recall_score(y_true, y_pred, average='macro')
    f1 = f1_score(y_true, y_pred, average='macro')

    print(f"Precision: {precision}, Recall: {recall}, F1-Score: {f1}")

# Example usage
evaluate_model(model, test_dataset)
```

Figure-6: Evaluation Metrics

3.5.Experimental Setup

3.5.1.Hardware Configuration

To attain efficient training along with inference capability, the research team is using NVIDIA A100 or RTX 3090 high performance GPUs. CPUs that keep up fast operation speeds and sufficient memory storage are needed in processing tasks.

3.5.2.Software Configuration

Libraries such as OpenCV and NumPy support data preprocessing tasks as well as augmentation tasks. The application that practitioners use transfer learning models is from PyTorch Model Zoo and TensorFlow Hub. A robust and efficient object detection system based on sequential data analytics typically combined with selection techniques and multiple activation functions together with superior architectural constructs is proposed.

```
# Check GPU Availability
gpus = tf.config.experimental.list_physical_devices('GPU')
if gpus:
    print("GPU is available")
else:
    print("GPU is not available")
```

Figure-7: Software Configuration.

4.DISCUSSION

4.1.Interpretation of Results

The proposed model uses a combination of CNN and LSTM which provides better results than standard methods and in particular, when applied in sequential verification scenarios such as video object identification. This framework integrates CNNs to extract spatial features and LSTMs to model temporal dependencies in order to improve both precision and stability of analysis results.

4.1.1.Why the Proposed Model Outperforms Baseline Methods

Spatial Feature Extraction: CNN component performs object identification within separate frames through its edge and shape and texture analysis capabilities. Successful detection can be maintained through CNN's design format across complex visual conditions.

Temporal Dependency Modeling: The model is based on its LSTM component, which is used to process sequences to identify the temporal changes between objects. The mentioned technology is suitable to process scenarios with occluded or rapidly moving objects and moving backgrounds.

Attention Mechanism: Frame sequence filteration enhances model performance via the attention mechanism that cuts out detection noise without harming detection consistency.

4.1.2.Role of LSTMs in Improving Accuracy

Video object detection applications become important for LSTMs to be able to capture distant dependencies in sequential information. Through its framework in which LSTMS are used to keep track of past frames, the model is able to effectively predict trajectories and handle temporal variation. The LSTM analytics establish a position map of moving cars from video sequences, even when these vehicles are hidden behind blocking objects. In contrast, when using this model the detection accuracy significantly improves when compared to CNN based systems that work alone.

Code to Visualize LSTM Outputs:

```
import matplotlib.pyplot as plt

# Extract LSTM outputs for visualization
sample_input = next(iter(test_dataset))[0]  # Get a sample batch from the test dataset
lstm_outputs = lstm_model(sample_input)  # Pass through the LSTM model

# Plot LSTM outputs
plt.figure(figsize=(10, 6))
for i in range(5):  # Plot outputs for the first 5 sequences
    plt.plot(lstm_outputs[i], label=f'Sequence {i+1}')
plt.title("LSTM Outputs Over Time")
plt.xlabel("Time Steps")
plt.ylabel("Output Values")
plt.legend()
plt.show()
```

Figure-8: Code to Visualize LSTM Outputs.

5.CONCLUSION

To address this challenge, this work proposes a spatial features extraction and temporal dependency modeling hybrid CNN-LSTM framework for object detection in sequential data. The proposed model employs CNNs to detect objects within single frames and LSTMs to model temporal relationships between frames, outperforming these methods alone. The work includes key contributions of incorporating an attention mechanism to highlight important frames and showing improved performance on benchmark datasets. When understanding object movement and context is crucial to the dynamic environment like video analysis, the hybrid approach works extremely well. The model holds promise, but will continue to be challenged by computational complexity and dataset limitations. Finally, future work may take the form of transformer based architectures and lightweight designs, to enable better efficiency and scalability. We explore the necessity of combining spatial and temporal models to analyze sequential data in this research; this opens the door to improvement in applications such as autonomous driving, surveillance, or some type of healthcare.

REFERENCE

1. H. Yin, "Object detection based on deep learning," *Transactions on Computer Science and Intelligent Systems Research*, vol. 5, pp. 547–555, Aug. 2024, doi: 10.62051/bcefz924.

2. H. Liu, "Dual attention-enhanced SSD: A novel deep learning model for object detection," *Applied and Computational Engineering*, vol. 57, no. 1, pp. 26–39, Apr. 2024, doi: 10.54254/2755-2721/57/20241308.

3. R. A. Dakhil and A. R. H. Khayeat, "Deep Learning for Enhanced Marine Vision: Object detection in underwater environments," *International Journal of Electrical and Electronics Research*, vol. 11, no. 4, pp. 1209–1218, Dec. 2023, doi: 10.37391/ijeer.110443.

4. A. M. Reda, N. El-Sheimy, and A. Moussa, "DEEP LEARNING FOR OBJECT DETECTION USING RADAR DATA," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-1/W1-2023, pp. 657–664, Dec. 2023, doi: 10.5194/isprs-annals-x-1-w1-2023-657-2023.

5. P. Thangavel and S. Karuppannan, "Dynamic Event Camera Object Detection and Classification Using Enhanced YOLO Deep Learning Architectur," *Optica OPen*, Nov. 2023, doi: 10.1364/opticaopen.24532636.v1.

6. I. García-Aguilar, J. García-González, R. M. Luque-Baena, and E. López-Rubio, "Object detection in traffic videos: an optimized approach using super-resolution and maximal clique algorithm," *Neural Computing and Applications*, vol. 35, no. 26, pp. 18999–19013, Jun. 2023, doi: 10.1007/s00521-023-08741-4.

7. S. A. Nawaz, J. Li, U. A. Bhatti, M. U. Shoukat, and R. M. Ahmad, "AI-based object detection latest trends in remote sensing, multimedia and agriculture applications," *Frontiers in Plant Science*, vol. 13, Nov. 2022, doi: 10.3389/fpls.2022.1041514.

8. B. Hwang, S. Lee, and H. Han, "LNFCOS: Efficient Object Detection through Deep Learning Based on LNblock," *Electronics*, vol. 11, no. 17, p. 2783, Sep. 2022, doi: 10.3390/electronics11172783.

9. H.-J. Park, J.-W. Kang, and B.-G. Kim, "SSFPN: Scale Sequence (S2) Feature-Based Feature Pyramid Network for object detection," *Sensors*, vol. 23, no. 9, p. 4432, Apr. 2023, doi: 10.3390/s23094432.

10. N. Kumar, "Object Detection using Artificial Intelligence and Machine Learning," *SciSpace - Paper*, Aug. 2021, [Online]. Available: https://typeset.io/papers/object-detection-using-artificial-intelligence-and-machine-3l22987dr3

11. I. G. B. Sampaio, L. Machaca, J. Viterbo, and J. Guérin, "A novel method for object detection using deep learning and CAD models.," *SciSpace - Paper*, Feb. 2021, [Online]. Available: https://typeset.io/papers/a-novel-method-for-object-detection-using-deep-learning-and-4tu99a9l43

12. S. Burde and S. V. Budihal, "Multiple object detection and tracking using deep learning," in *Lecture notes in electrical engineering*, 2021, pp. 257–263. doi: 10.1007/978-981-33-4866-0_32.

13. D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, Apr. 2020, doi: 10.1186/s13673-020-00219-9.

14. A. B. Qasim and A. Pettirsch, "Recurrent Neural Networks for video object detection.," *SciSpace - Paper*, Oct. 2020, [Online]. Available: https://typeset.io/papers/recurrent-neural-networks-for-video-object-detection-4drpghk9oc

15. B.-Y. Chen, Y.-K. Shen, and K. Sun, "Research on object detection algorithm based on multilayer information fusion," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–13, Sep. 2020, doi: 10.1155/2020/9076857.

16. R. Huang *et al.*, "An LSTM approach to temporal 3D object detection in LiDAR point clouds," in *Lecture notes in computer science*, 2020, pp. 266–282. doi: 10.1007/978-3-030-58523-5_16.

17. A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, Dec. 2019, doi: 10.1016/j.inffus.2019.12.012.

18. A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artificial Intelligence Review, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/s10462-020-09825-6.

19. F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," Sensors, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/s16010115.

20. Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the last mile of artificial intelligence with edge computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, Jun. 2019, doi: 10.1109/jproc.2019.2918951.