iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85503

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

Recipe Generation from Food Image Using Deep Learning

Vijaya Durga H[1], Yerram Shiva Nagi Reddy[2], Shree Vishun J[3]
Department of Computer Science and Engineering Ballari Institute of Technology and Management (BITM), Ballari, India

Abstract: ***Seeing a dish and wanting to cook it is a common experience, but a photograph alone does not tell us what went into the food or how it was made. This paper presents a web-based system that identifies a dish from a single food photograph and returns a ready-to-follow recipe. A deep convolutional network trained on the Food-101 dataset classifies the uploaded image into one of 101 dish categories and reports the five most likely candidates with confidence scores. The recognised label is then used to fetch a structured recipe from a curated recipe knowledge base, consisting of an ingredient checklist with quantities, ordered cooking instructions, cuisine, preparation time, servings and difficulty. The result is shown in a browser interface that also lets the user download the recipe as a text file or print it. On a test pizza image, the system identified the dish with a confidence of 99.7% and returned a nine-ingredient, seven-step recipe while running on a CPU-only machine. The paper describes the design, implementation and observed behaviour of the system, discusses its limitations, and outlines how it can be extended toward true image-conditioned recipe generation.***

Index Terms: ***Recipe generation, food recognition, deep learn-ing, convolutional neural networks, Food-101, image classifica-tion, web application***

I. INTRODUCTION

Food photography has become a routine part of everyday life. Millions of dishes are photographed and shared online every day, yet a photograph by itself says very little about how a dish was prepared. A person who sees an appealing dish at a restaurant or on social media usually has no direct way of learning what went into it or how to cook it at home. Recovering a recipe from an image is a natural problem for computer vision, but it is considerably harder than ordinary image classification. A recipe is a structured object made up of a title, a set of ingredients and an ordered list of instructions. Many ingredients, such as salt, oil, spices or the milk and eggs inside a batter, leave little or no visual trace. The same dish can also look very different depending on lighting, plating and camera angle, while different dishes can look almost identical. Deep convolutional neural networks (CNNs) have reached strong performance on food classification benchmarks [6], and large paired image–recipe datasets [2] have encouraged research on full recipe understanding. Fully generative ap-proaches [3], however, are heavy to train and to deploy. For a practical tool that an ordinary user can run on a laptop, a lighter design is attractive: recognise the dish reliably, and then present a well-written recipe for it.

This paper follows that design. The main contributions are:

1) A complete image-to-recipe pipeline that combines a deep learning dish classifier with a structured recipe knowledge base.
2) A browser-based interface that shows the recognised dish, a top-5 confidence breakdown, an ingredient checklist and numbered cooking steps, with options to download or print the result.
3) A working demonstration on a CPU-only machine, to-gether with a discussion of its limitations and of the steps required to move toward image-conditioned recipe generation.

The rest of the paper is organised as follows. Section II reviews related work, Section III describes the methodology and implementation, Section IV presents and discusses the results, and Section V concludes the paper. Section VI outlines future work.

II. LITERATURE REVIEW

A. Food Recognition and Ingredient Prediction

Early work in food computing concentrated on recognising the category of a dish. Bossard *et al.* introduced the Food-101 dataset and showed that random forests over image patches could classify 101 food categories with moderate accuracy [6]. Once deep CNNs became widely available, accuracy on such benchmarks improved sharply. Chen and Ngo went beyond dish categories and proposed a multi-task network that recognises ingredients directly and uses them to retrieve matching recipes [7]. A broad survey of the field, covering recognition, recommendation and health-oriented applications, is given by Min *et al.* [8].

©IJRASET: All Rights are Reserved

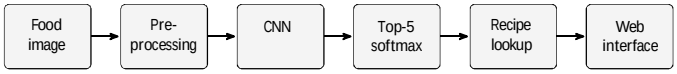


Fig. 1. Overall pipeline of the proposed system: from an uploaded photograph to a displayed recipe.

B. *Cross-Modal Recipe Retrieval*

A second line of work treats the problem as retrieval. Salvador *et al.* learned a joint embedding space in which food images and recipe texts lie close together when they match, enabling image-to-recipe search [1]. This was supported by the Recipe1M dataset, later extended to Recipe1M+ with more than one million recipes and 13 million images [2]. Retrieval methods produce well-formed recipes, but they are limited by the contents of the database.

C. *Recipe Generation*

Generation removes this limitation. Salvador *et al.* proposed the *inverse cooking* system, which predicts ingredients as a set and then generates instructions with a Transformer conditioned on the image and the ingredients [3]. Transformer decoders [5] have largely replaced recurrent models such as the LSTM [11] for this kind of text generation, because they handle long-range dependencies better. Such models, however, need large paired datasets and substantial compute.

D. *Backbone Networks*

Residual networks [4] and EfficientNet [9] are widely used visual backbones, and Vision Transformers [10] have become a strong alternative. Any of these can serve as the recognition stage of a system like the one described here.

III. METHODOLOGY

A. *Dataset*

The classifier is trained on Food-101 [6], which contains 101 dish categories with 1,000 images per category, split into 750 training and 250 test images per class. The images are collected from the web, so they include natural variation in lighting, viewpoint and background. Because the app is built on this dataset, it recognises exactly these 101 categories and no others.

B. *System Overview*

Fig. 1 shows the pipeline. The user uploads a food image through the browser. The backend pre-processes the image and passes it to the CNN classifier, which outputs a probability for each of the 101 classes. The top-five classes are returned with their confidence scores, and the highest-ranked label is used as a key into the recipe knowledge base. The matching recipe, along with its metadata, is sent back to the browser and rendered.

C. *Image Pre-processing*

Uploaded images (JPEG, PNG, BMP or WEBP, up to 10 MB) are converted to RGB, resized to the input resolution of the network, and normalised using the same statistics that were applied during training. [TBD: state input size, e.g., 224×224, and normalisation values]. During training, standard augmentation such as random cropping and horizontal flipping was applied [TBD: confirm augmentations used].

D. *Dish Classifier*

The recognition stage is a deep convolutional network [TBD: name the backbone, e.g., ResNet-50 [4] or EfficientNet [9], and whether it was initialised from pretrained weights] fine-tuned on Food-101. For an input image $\mathbf{x}$, the network produces logits $\mathbf{z} \in \mathrm{R}^{101}$, which are converted to class probabilities by the softmax function:

$$p_k = \frac{\exp(z_k)}{\sum_{j=1}^{101} \exp(z_j)}, \quad k = 1, \ldots, 101. \qquad (1)$$

Training minimises the categorical cross-entropy

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{101} y_{ik} \log p_{ik}, \qquad (2)$$

where yik is the one-hot ground-truth label of sample i. [TBD: state optimiser, e.g., Adam [12], learning rate, batch size and number of epochs].

At inference, the predicted dish is $c\hat{} = \arg\max_k p_k$, and the five highest probabilities are reported to the user. Showing the top-five list, rather than only the best guess, makes the system more transparent: a user can see at a glance when the model is unsure and when several dishes are plausible.

E. Recipe Knowledge Base

Each of the 101 dish labels is linked to a structured recipe entry containing the following fields: cuisine, total time, servings, difficulty level, an ingredient list with quantities, and an ordered list of cooking instructions. When a dish is recognised, the entry for the predicted label is fetched and returned. Keeping the recipes in a separate store, rather than inside the network, means that recipes can be corrected or extended without retraining the classifier.

F. Implementation

The system is implemented as a web application served locally (at 127.0.0.1:5000 during development). The backend loads the trained model once at start-up and shows a "Model ready (CPU)" status badge in the interface, confirming that inference runs on a CPU without any GPU. The front end supports click-to-choose and drag-and-drop upload, shows a preview of the chosen image, and provides *Identify food & generate recipe* and *Clear* buttons. The final recipe can be downloaded as a .txtfile or printed directly from the page.

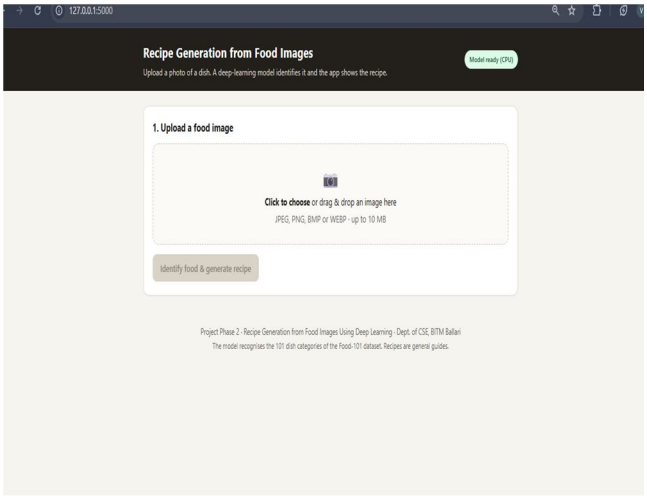


Fig. 2. Start page of the application, showing the upload area and the model status indicator

TABLE I
SAMPLE OUTPUT FOR THE TEST PIZZA IMAGE

Field	Value
Identified dish	Pizza
Top-1 confidence	99.7%
Cuisine	Italian
Total time	1 hr 30 min
Servings	2 pizzas
Difficulty	Medium
Ingredients listed	9
Instruction steps	7

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

IV. RESULTS AND DISCUSSION

A. User Interface

Fig. 2 shows the start page of the application. The upload area accepts common image formats, and the main action button stays disabled until an image has been selected, which prevents empty requests. A footer clarifies that the model recognises the 101 Food-101 categories and that the recipes are general guides.

B. Image Upload and Recognition

Fig. 3 shows the interface after a pizza photograph has been uploaded. The preview lets the user confirm that the correct file was chosen before running recognition.

The recognition result for this image is shown in Fig. 4. The system identifies the dish as *Pizza* with a confidence of 99.7%, and the remaining four candidates (bruschetta, garlic bread, omelette and beef carpaccio) each receive a probability close to zero. The result card also shows the recipe metadata drawn from the knowledge base, which is summarised in Table I.

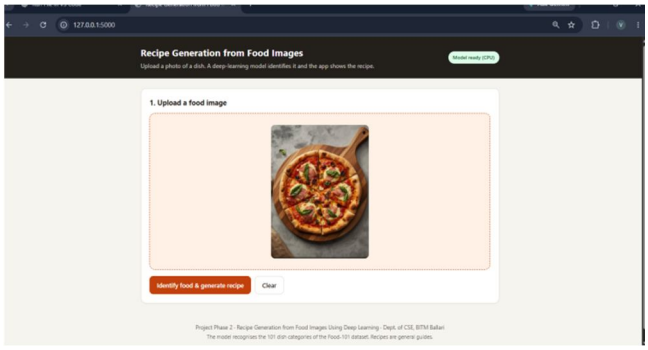


Fig. 3. Interface after uploading a test food image (a pizza).

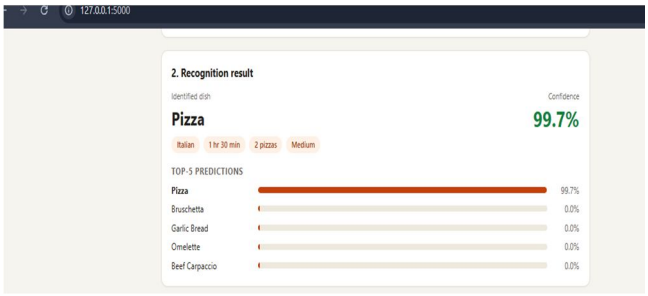


Fig. 4. Recognition result with the identified dish, confidence score and top-5 predictions.

C. Generated Recipe

Fig. 5 shows the recipe returned for the pizza image. The ingredient panel lists flour, yeast, salt, warm water, olive oil, tomato sauce, mozzarella, fresh basil and toppings of the user's choice, each with a check box so that the list can be used while shopping or cooking. The instruction panel presents seven numbered steps: kneading the dough, letting it rise, preheating

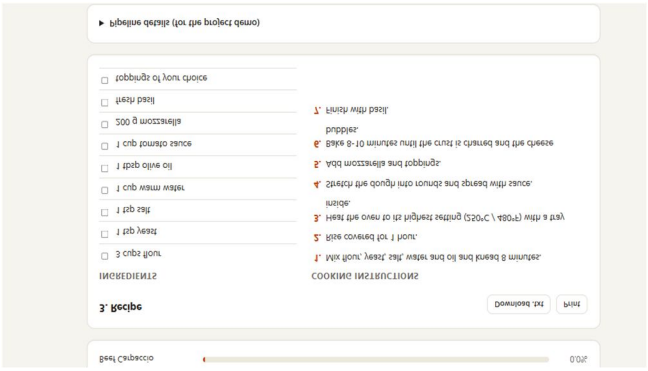


Fig. 5. Recipe output with an ingredient checklist, numbered cooking instructions, and download and print options.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLE II
CLASSIFIER PERFORMANCE ON THE FOOD-101 TEST SPLIT

Metric	Value
Top-1 accuracy Top-5 accuracy	[TBD: value]
Inference time per image (CPU)	[TBD: value]
	[TBD: value]

the oven with a tray inside, stretching and saucing the dough, adding cheese and toppings, baking until the crust is charred, and finishing with basil. The steps follow a sensible order and include practical details such as oven temperature and baking time.

D. Quantitative Evaluation

Table II lists the evaluation figures for the classifier on the Food-101 test split. Top-1 accuracy is the fraction of test images whose highest-probability class is correct, and top-5 accuracy is the fraction whose correct class appears among the five highest-probability classes:

$$\mathrm{Acc}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[c_i \in \mathrm{Top}_k(\mathbf{p}_i)\right]. \qquad (3)$$

E. Limitations

Several limitations should be stated openly. First, the sys-tem recognises only the 101 Food-101 categories, so dishes outside this set, including many regional Indian dishes, will be assigned to the nearest known class. Second, the recipe is attached to the predicted label rather than generated from the specific image, so two different pizzas will receive the same recipe, and the recipe cannot reflect visible variations such as extra toppings. Third, the system assumes one dominant dish per photograph and may be unreliable for mixed plates. Fi-nally, the recipes are general guides, and allergen information must not be inferred from them, since the actual ingredients of a photographed dish may differ.

V. CONCLUSION

This paper presented a web-based system that identifies a dish from a food photograph using a deep convolutional net-work trained on Food-101 and returns a structured recipe with ingredients, instructions and metadata. The implementation runs on a CPU-only machine, and a test with a pizza image produced a confident prediction and a complete, usable recipe. The design keeps recognition and recipe content separate, which makes the recipes easy to edit without retraining. The main limitations are the fixed set of 101 dishes and the fact that recipes are retrieved rather than generated from the image itself.

VI. FUTURE SCOPE

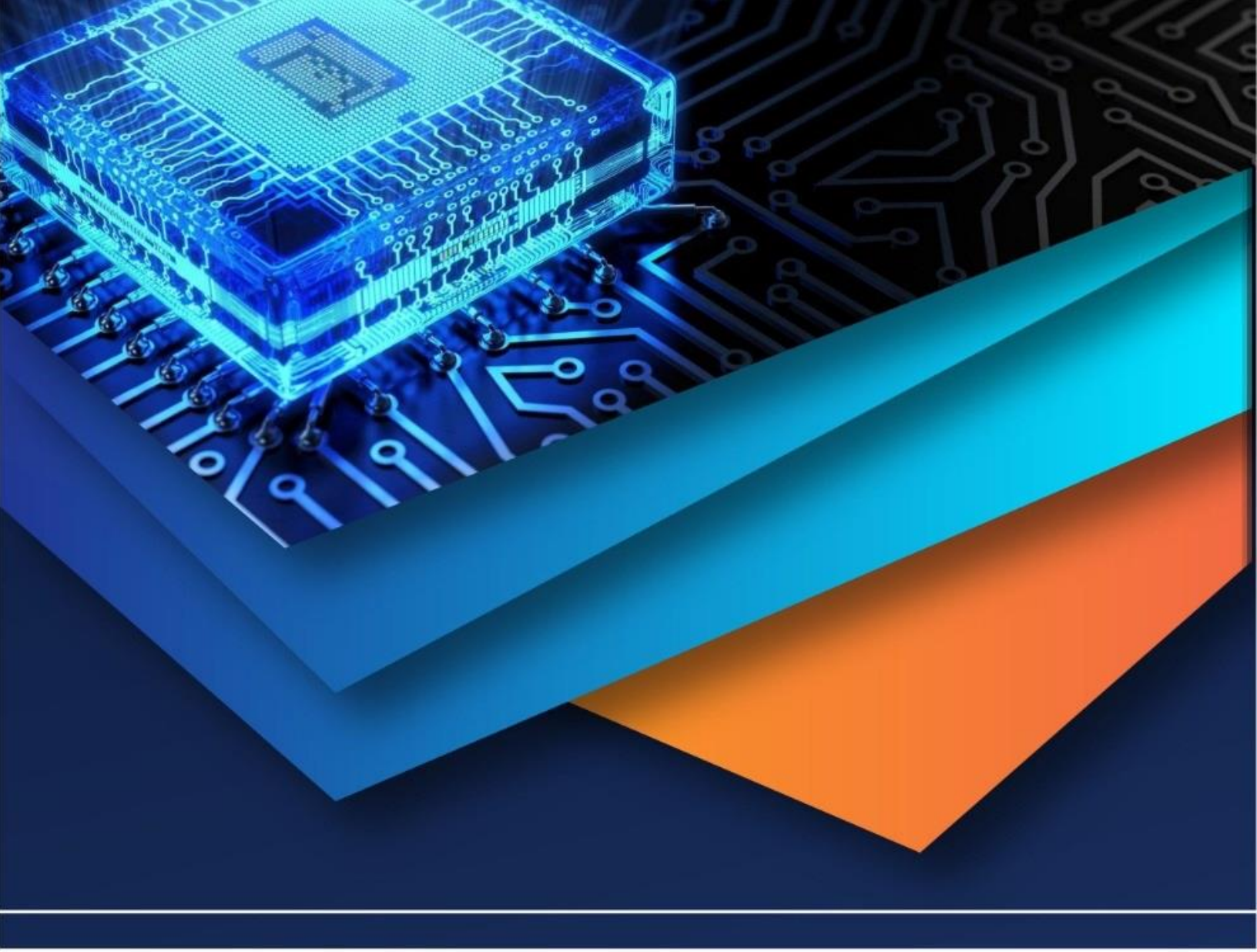
The most direct extension is to replace the lookup stage with a learned generator that predicts ingredients from the image and produces instructions with a Transformer decoder [5], as in inverse cooking [3]; such a system could be evaluated with text-overlap metrics such as BLEU [13]. Other improvements include a stronger backbone such as a Vision Transformer [10], extending the dish list with regional Indian cuisines, estimating portion sizes and nutritional values, supporting dietary filters (vegetarian, allergen-free), and deploying the model on mobile devices.

REFERENCES

[1] A. Salvador et al., “Learning cross-modal embeddings for cooking recipes and food images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, 2017, pp. 3020–3028.
[2] J. Marin et al., “Recipe1M+: A dataset for learning cross-modal embed-dings for cooking recipes and food images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 187–203, Jan. 2021.
[3] A. Salvador, M. Drozdzal, X. Giro-i-Nieto, and A. Romero, “Inverse cooking: Recipe generation from food images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, 2019, pp. 10453–10462.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.
[5] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Long Beach, CA, USA, 2017, pp. 5998–6008.
[6] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining discriminative components with random forests," in Proc. Eur. Conf. Comput. Vis. (ECCV), Zurich, Switzerland, 2014, pp. 446–461.
[7] J. Chen and C.-W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in Proc. ACM Int. Conf. Multimedia, Amsterdam, The Netherlands, 2016, pp. 32–41.
[8] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," ACM Comput. Surv., vol. 52, no. 5, pp. 1–36, Sep. 2019.
[9] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for con-volutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), Long Beach, CA, USA, 2019, pp. 6105–6114.
[10] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015.
[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL), Philadelphia, PA, USA, 2002, pp. 311–318.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)