



Iliyas Makhatbek ^{1,*} , Symbat Kabdrakhova ² 

¹Kazakh British Technical University, Almaty, Kazakhstan

²Al-Farabi Kazakh National University, Almaty, Kazakhstan

*e-mail: il_makhatbek@kbtu.kz

KAZAKH TRADITIONAL FOOD IMAGE CLASSIFICATION USING CNNs

Abstract. Obesity is becoming an increasingly serious global health issue with severe consequences. Effective nutrition management is crucial in combating this epidemic. In Kazakhstan, traditional foods are a central part of the culture, yet comprehensive data and tools for analyzing dietary habits are lacking. Leveraging advances in computer vision, we developed a convolutional neural network (CNN) based approach to automatically classify images of traditional Kazakh dishes. We compiled a new dataset of 9,577 images across 22 categories of Kazakh foods and used it to train and evaluate several CNN models. The best-performing model (a fine-tuned DenseNet121) achieved a top-1 classification accuracy of 95%. These results highlight the potential of AI-based food recognition for dietary monitoring, nutritional assessment, and cultural preservation. Furthermore, the trained model was deployed in a Telegram chatbot to enable real-time food identification through image uploads, demonstrating a practical application of the system.

Key words: Kazakh cuisine, food image classification, convolutional neural networks, transfer learning, computer vision, dietary monitoring.

1. Introduction

The saying “You are what you eat” reflects the strong connection between diet and overall health. In recent years, concerns about nutrition and obesity have grown worldwide. According to the World Health Organization (WHO), 21% of children in Kazakhstan aged 6–9 are overweight or obese, linking their diets high in calories, sugar, and fat to serious health issues such as heart disease and type 2 diabetes [1]. One way to maintain a healthy lifestyle is to monitor food intake, but manually logging meals and counting calories is time-consuming and prone to errors. Advances in computer vision (CV) offer a more efficient alternative—food recognition technology that can automatically identify dishes from images. This technology has promising applications in dietary monitoring and nutritional assessment, restaurant automation, food quality control, and even assistance for visually impaired individuals [2].

A variety of food image datasets have been compiled to support CNN training, though most focus

on Western and East Asian cuisines. (see Table 1). For example, Food-101 is a widely used dataset featuring 101 Western food categories with 1,000 images each [3]. UECFOOD256 covers 256 Japanese dishes [4], and ChineseFoodNet includes 208 categories of Chinese cuisine [5]. Larger datasets like Food2K span 2,000 food classes worldwide (over 1 million images) but remain private [6]. Similarly, FoodX-251 provides 251 international food categories (158,000 images) [7], and ISIA Food-500 expands to 500 classes (nearly 400,000 images) across diverse global cuisines [8]. A recent effort focusing on regional foods is the Central Asian Food Dataset (CAFD) [9], which includes 42 classes of Central Asian dishes. However, CAFD, while valuable, primarily covers a few popular Kazakh dishes (like beshbarmak and kazy) and lacks many traditional foods and regional variants crucial for a comprehensive representation of Kazakh cuisine. In general, there is a clear gap in datasets and tools dedicated to Kazakh traditional foods, which are an important part of the country’s cultural heritage.

Table 1 – Summary of food classification datasets.

Dataset Name	Year	#Classes	#Images	Cuisine Focus
UECFood256	2014	256	31,000+	Japanese
ChineseFoodNet	2017	208	180,000+	Chinese
FoodX-251	2019	251	158,000+	International
ISIA Food-500	2020	500	399,726	International
Food2K	2021	2000	1,036,564	International
Food-101	2022	101	101,000	Western
CAFD	2022	42	16,499	Central Asian

To achieve high accuracy, most food classification studies leverage transfer learning with modern CNN architectures pre-trained on large general image databases. Models such as VGG-16 [10], ResNet-50 [11], MobileNetV2 [12], EfficientNet [13], and DenseNet [14] have all been successfully fine-tuned for food image recognition. By starting from models pre-trained on ImageNet [15], researchers can exploit useful generic features (edges, textures, shapes) and then adapt them to the specific nuances of food imagery. This approach addresses the limited samples per class in many food datasets and has consistently produced state-of-the-art results. For instance, an ensemble of fine-tuned CNN models (combining ResNet50, VGG19, MobileNetV2 and others) achieved over 96% accuracy on the Food-11 dataset by using transfer learning and model fusion [16]. With careful architecture selection and fine-tuning, even single CNN models now surpass 95% top-1 accuracy on challenging benchmarks like Food-101 [17]. Notably, researchers have demonstrated that multiple CNNs can be deployed together to improve robustness – for example, a smartphone-based food recognition system combined several deep CNNs to attain high accuracy in real time [18]. These advances illustrate the effectiveness of deep CNNs and transfer learning for recognizing food items from images. Beyond standard CNN classifiers, recent work has explored techniques to better handle the fine-grained nature of food recognition. One strategy is to incorporate attention mechanisms or part-based feature learning to focus on the most discriminative regions of the food image. For example, Min et al. [8] employed a stacked global-local attention CNN to improve recognition on the ISIA Food-500 dataset, enabling the model to zoom in on subtle details (like specific garnishes or textures) that distinguish similar dishes. Feng et al. [19] proposed a fine-grained recognition

method for Chinese cuisine that explicitly identifies important parts of the dish (such as certain ingredients or shapes), which boosted classification performance on visually similar food categories. Another emerging direction is the integration of vision transformers (ViTs) with CNNs to capture long-range dependencies in food images. By combining a CNN backbone for local feature extraction with transformer-based global context modeling, hybrid models have achieved further accuracy improvements on diverse food datasets [20]. These advanced architectures help address cases where purely local features are insufficient – for instance, distinguishing two soups might require attention to the overall arrangement of ingredients, which transformers can provide. Overall, the introduction of attention and transformer modules has enhanced CNN-based food classifiers, enabling them to better handle the inherent fine-grained complexity of food images.

Despite the progress in food image classification, prior research has largely overlooked Kazakh traditional cuisine. To bridge this gap, we present the first large-scale image dataset devoted to traditional Kazakh cuisine and use it to develop a robust food classification model. The new Kazakh Food Dataset contains 9,577 images spanning 22 distinct dish categories, encompassing a broad variety of local foods from meats and soups to dairy products and breads. Using this dataset, we trained and fine-tuned several state-of-the-art CNN architectures (ResNet50, EfficientNet-B0, VGG16, MobileNetV2, and DenseNet121) via transfer learning to determine the most effective model for recognizing Kazakh dishes. The goal of this research is not only to achieve high classification accuracy but also to demonstrate real-world utility. To that end, we integrated our best model into a Telegram messaging bot, allowing users to classify dishes by simply uploading a photo. This makes the technology acces-

sible and practical for everyday use, aiding dietary tracking and showcasing AI for cultural preservation.

In summary, the contributions of this work include:

- **New Dataset:** We compiled a novel dataset of Kazakh traditional food images (9,577 images, 22 categories), addressing an underrepresented domain in food recognition research.

- **Model Evaluation:** We applied and compared five modern CNN models for food image classification, achieving up to 95% accuracy with the best model (DenseNet121). We also analyzed the models' performance to understand the impact of architecture on this task.

- **Practical Deployment:** We deployed the top-performing model as a Telegram bot for real-time food image classification, illustrating the practical feasibility of our approach for end-users.

2. Materials and Methods

The primary objective of this study was to develop an accurate and reliable model for classifying images of traditional Kazakh foods. Figure 1 provides an overview of the methodological pipeline, from dataset creation to model training and deployment. The following subsections detail the dataset, preprocessing steps, CNN architectures, training procedure, and evaluation metrics.

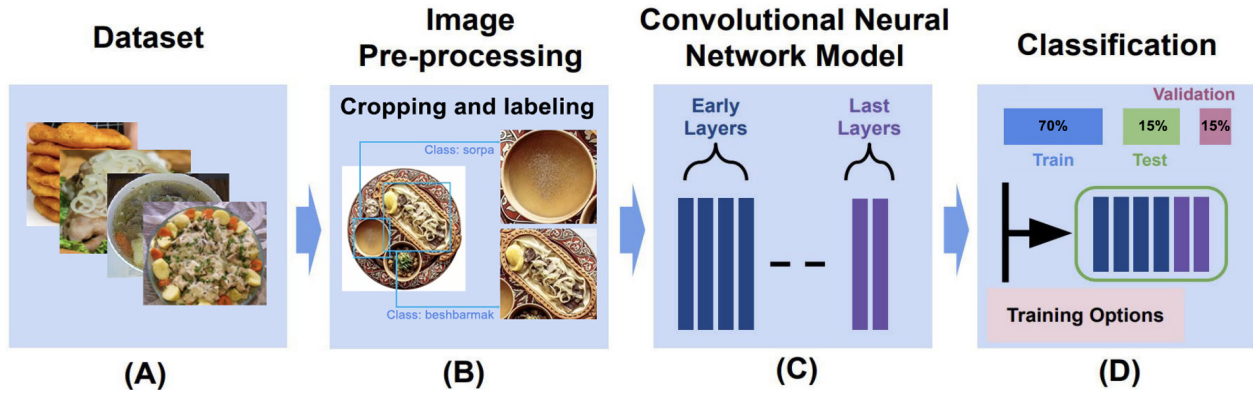


Figure 1 – (A) Collection of Kazakh food images. (B) Pre-processing images to normalize input dimensions, enhance feature extraction, and minimize overfitting. (C) Incorporation of pretrained convolutional neural networks with customized classifier layers. (D) Initial training, followed by fine-tuning to optimize model performance.

2.1. Dataset Collection

We created a dataset specifically to capture the diversity of traditional Kazakh cuisine. Images were sourced from public domains, including search engines (e.g., Google, Yandex) and social media platforms where people share photographs of local dishes (Figure 2). To increase intra-class variability in lighting, angle, and background conditions, additional frames were extracted from public YouTube cooking videos featuring traditional Kazakh dishes. This approach allowed us to partially simulate real-world conditions and reduce dataset bias. However, no photographs were taken directly under field conditions by the authors, which remains a limitation for further expansion. To ensure quality, an initial manual filtering removed duplicates, very low-resolution images, and irrelevant content (such as non-food items or incorrect labels). The resulting

dataset comprises 9,577 images categorized into 22 distinct Kazakh food classes. These classes include well-known dishes like beshbarmak (boiled meat with noodles) and plov (rice pilaf), as well as less internationally known items such as boursak (fried bread), qazy-qarta (horsemeat sausage and cured fat), samsa (meat pastry), sorpa (meat broth soup), qurt (dried cheese curds), and many others. The number of images per class ranges from 96 to 920 (Figure 3), reflecting some natural frequency imbalance in available data. To enable unbiased model evaluation, we partitioned the dataset into training (70% of images), validation (15%), and test (15%) subsets. The split was stratified by class so that all classes are represented in each subset. This balanced split ensures that model performance is assessed on food images it has never seen during training.



Figure 2 – Sample images from the Kazakh Traditional Food Dataset.

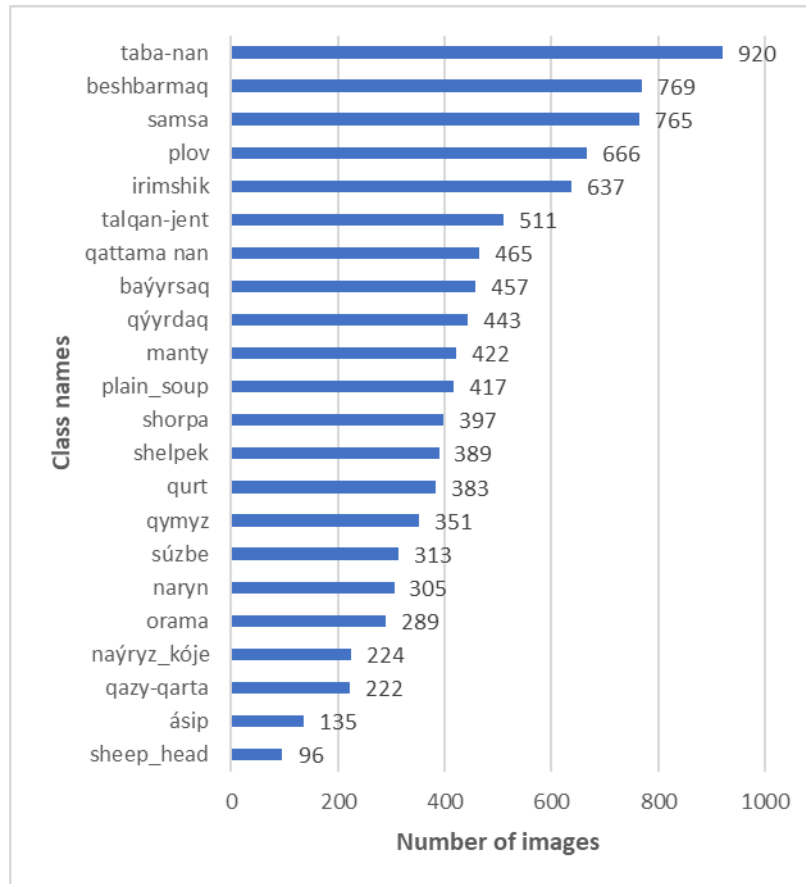


Figure 3 – Image distribution across Kazakh food categories.

2.2. Image Preprocessing and Augmentation

Prior to training the CNN models, we applied a series of preprocessing and augmentation steps to standardize the data and improve generalization:

Resizing: Each image was resized to 224×224 pixels, matching the input dimension requirements of the CNN architectures used. This uniform size ensures compatibility with models like ResNet50, VGG16, etc., which typically expect 224×224 RGB inputs.

Normalization: We normalized pixel intensities based on the mean and standard deviation of the ImageNet dataset (on which our CNN backbones were pre-trained). This scaling (subtracting mean and dividing by std for each color channel) brings the image data into a range suitable for the pre-trained models and stabilizes training.

Data Augmentation: To increase effective dataset size and help the model become invariant to various transformations, we applied random augmentations during training. These included random rotations (up to ~ 15 degrees), horizontal flips, adjustments to brightness and contrast, and slight translations or zooms. Augmentation introduces variability, reducing the chance of overfitting and improving the model's ability to handle real-world image variation (different angles, lighting, backgrounds, etc.).

Class Imbalance Mitigation: For underrepresented classes (those with relatively fewer images), we employed targeted augmentation. Specifically, images from these minority classes were augmented more heavily or more frequently to synthetically boost their presence during training. This strategy helps the model not to be biased in favor of classes with many examples. By augmenting *ásip* (a type of sausage) or *shelpek* (flatbread), for instance, which had fewer original samples, we help the model learn those classes nearly as well as classes like *beshbarmaq* which had abundant examples.

2.3. CNN Architectures and Transfer Learning

Deep convolutional neural networks (CNNs) have achieved state-of-the-art results in image recognition tasks and are well-suited for food classification [2]. In this work, we leveraged transfer learning: instead of training from scratch, we fine-tuned models pre-trained on the large ImageNet dataset [15] for our specific food classification task. Transfer learning capitalizes on the generic visual features (edges, textures, shapes) learned from millions of generic images and adapts them to our domain (Kazakh food images), which is particularly effective given our dataset's moderate size. We selected

five well-known CNN architectures, chosen to provide a mix of depth, parameter size, and design philosophies:

ResNet50 [11] – a 50-layer deep residual network. ResNet introduced skip connections (residual links) that help propagate gradients and mitigate the vanishing gradient problem in very deep networks [11]. ResNet50 is capable of extracting rich features through its many layers, yet trains effectively due to these identity connections.

EfficientNet-B0 [13] – a model from the EfficientNet family that optimizes accuracy per parameter by balancing network depth, width, and resolution [13]. EfficientNet-B0 is a relatively lightweight model (~ 5.3 M parameters) but achieves high accuracy by a compound-scaling strategy, making it an excellent choice when computational efficiency is a concern.

VGG16 [10] – a classic 16-layer CNN known for its simple sequential architecture of convolutional layers with small 3×3 filters [10]. VGG16 has a large number of parameters (~ 138 M) and was among the first very deep networks to show outstanding performance on ImageNet. Its hierarchical feature learning is effective, though the lack of residual connections and high parameter count can make training and fine-tuning slower or prone to overfitting.

MobileNetV2 [12] – a convolutional network architecture designed for mobile and embedded devices. MobileNetV2 uses inverted residual blocks and depthwise separable convolutions to drastically reduce computation and model size, while still achieving competitive accuracy [12]. It has around 3.4M parameters, making it the smallest of the models we tested. This model provides insight into how a lightweight architecture performs on our task.

DenseNet121 [14] – a 121-layer densely connected network. DenseNet connects each layer to every subsequent layer (feature maps are cumulatively reused), leading to strong feature propagation and efficiency [14]. With roughly 8 million parameters, DenseNet121 encourages feature reuse and mitigates vanishing gradients even with great depth. This architecture often yields very robust performance on limited datasets due to its feature reuse strategy.

Each model above was initialized with weights pre-trained on ImageNet's 1,000-class object recognition task [15]. We replaced each model's final fully connected classification layer with a new dense layer (with softmax activation) matching our 22 food categories. This customization allows the

network to output class probabilities for the Kazakh food classes. Initially, all convolutional layers retained the pre-trained weights (which capture general image features), and only the new final layer's weights were randomized for training.

2.4. Training Procedure

We trained all models using a two-stage transfer learning approach to gradually adapt the pre-trained networks to our specific dataset:

1) Initial Training (Feature Extraction Stage): In this phase, we froze all convolutional layers of the CNN (i.e., kept the pre-trained weights fixed) and trained only the newly added classifier layer (and optionally a few preceding fully connected layers, if present). This strategy limits the number of parameters being updated, preventing significant loss of learned features and requiring fewer training samples to converge. We used the Adam optimizer with a learning rate of 0.001 for fast convergence. A batch size of 32 was chosen, and training ran for 15 epochs in this stage. We employed the categorical cross-entropy loss function, appropriate for multi-class classification with softmax output. During this phase, the model learns to map the high-level features (extracted by the pre-trained base) to our specific classes. Early epochs saw rapid improvement in accuracy as the new layer adjusted to the task. We monitored performance on the validation set after each epoch, and used early stopping if validation accuracy plateaued to avoid overfitting.

2) Fine-Tuning (Full Network Training Stage): After the initial phase, the model's new classifier was reasonably well-trained while the convolutional base remained at its ImageNet-tuned state. In the fine-tuning stage, we unfroze some of the top convolutional layers of the network (typically the last few layers in the CNN backbone) and continued training at a much lower learning rate (we used 1×10^{-5}). Fine-tuning allows the model to adjust the more specific feature representations in deeper layers to better fit the nuances of Kazakh food imagery. We ran this stage for an additional 10 epochs, which was sufficient for the validation accuracy to stabilize. During fine-tuning, a small learning rate was crucial to avoid degrading previously learned features; it allowed for subtle adjustments. We found that fine-tuning improved performance primarily by increasing recall for classes that the feature extractor stage struggled with, thereby balancing the model across all categories.

All training was performed on a workstation with an NVIDIA GTX-series GPU. On average, each model's initial training stage took a couple of

hours, and fine-tuning took roughly one hour, although EfficientNet-B0 and MobileNetV2 (being smaller) trained faster than VGG16 and ResNet50. We saved the model with the highest validation accuracy for final evaluation on the test set.

Training Hyperparameters. The models were trained using the following settings:

- Initial learning rate: 0.001 (feature extraction stage), reduced to 1×10^{-5} during fine-tuning.
- Optimizer: Adam with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$).
- Loss function: Cross-entropy loss.
- Batch size: 32.
- Number of epochs: 15 for initial training, 10 for fine-tuning.
- Scheduler: StepLR with step size of 5 epochs and $\gamma = 0.1$ during fine-tuning.
- Weight decay: Not applied.

2.5. Evaluation Metrics

We evaluated model performance on the unseen test set using several standard classification metrics. The primary metric is accuracy, defined as the proportion of correctly classified images among all test images. If TP, TN, FP, and FN denote the counts of true positives, true negatives, false positives, and false negatives respectively, the accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

While accuracy gives an overall measure, it can be misleading if the test set is imbalanced among classes. Thus, we also compute precision and recall for each class, as well as their harmonic mean, the F1-score. For a given class (dish type) treated as the "positive" class, we have:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

which measures how many of the images that the model labeled as this class were actually that class. High precision means few false alarms (mislabeling other dishes as this one).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

which measures how many of the images of this class were correctly identified. High recall means the model misses very few images of that class.

The F1-score for a class is:

$$F1 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (4)$$

We report macro-averaged precision, recall, and F1, which are averages of these metrics across all 22 classes (giving each class equal weight). The macro metrics are informative in our case because they indicate how well the model performs on each dish category independently of class frequency. A high macro-F1 implies the model is consistently accurate across all dish types, not just the most common ones. We present the accuracy and macro metrics for each model, and also examine class-specific performance to identify which dishes are easiest or hardest to recognize.

3. Results

After training and fine-tuning the five CNN models, we evaluated each on the test set of Kazakh food images. Table 2 summarizes the performance of each model in terms of overall accuracy and macro-averaged precision, recall, and F1-score. All models achieved high accuracy above 89%, indicating that transfer learning on our dataset is effective. Among the architectures, DenseNet121 emerged as the top performer with 95% accuracy, slightly outperforming EfficientNet-B0 (94% accuracy) and clearly outperforming ResNet50, MobileNetV2, and VGG16. The DenseNet121 model also achieved the highest macro precision (95%) and F1-

score (95%), reflecting its strong and balanced performance across all dish categories. EfficientNet-B0 was a close second in most metrics, with a macro F1 of 93%. MobileNetV2 and ResNet50 reached about 91–92% accuracy and F1, while VGG16 trailed at 89% accuracy and a macro F1 of 88%.

As shown in Table 2, DenseNet121 achieved the highest accuracy on our dataset. Notably, after fine-tuning (FT), DenseNet121’s macro recall improved from 94% to 95%, indicating that fine-tuning helped it correctly capture a few additional instances of certain foods that were initially misclassified. The improvement in F1-score to 95% suggests a more balanced precision-recall trade-off after fine-tuning. In comparison, EfficientNet-B0 also performed very well, likely due to its excellent pre-training and efficient use of parameters, ending up only slightly behind DenseNet121 on all metrics. ResNet50 and MobileNetV2 had respectable accuracy (over 90%); ResNet50 benefited from its depth, while MobileNetV2’s lightweight design somewhat limited its ultimate accuracy but it still generalized well (its precision and recall both 92%). VGG16 had the lowest performance, which can be attributed to its very large number of parameters and lack of modern architectural features (no residual or dense connections). VGG16 tended to overfit slightly on the training data even with augmentation, and its macro recall (87%) was the lowest, indicating it struggled with some classes more than the others.

To visualize the training process, Figure 4 shows the learning curves of the DenseNet121 model.

Table 2 – Performance of different CNN models on the Kazakh Traditional Food test set. The DenseNet121 (Fine-tuned) model corresponds to the DenseNet after the second-stage training. All values are percentages (%).

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score	Notes
ResNet50	91%	92%	91%	91%	-
EfficientNet-B0	94%	94%	93%	93%	-
VGG16	89%	90%	87%	88%	-
MobileNet_v2	92%	92%	92%	92%	-
DenseNet121	95%	95%	94%	94%	Original version
DenseNet121 (FT)	95%	95%	95%	95%	Fine-tuned version

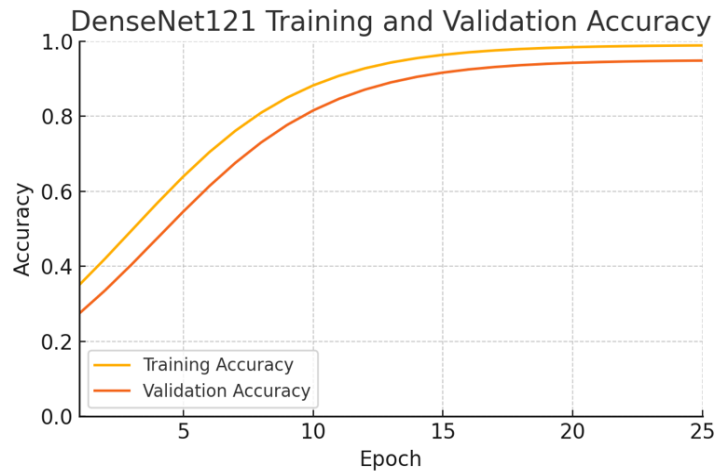


Figure 4 – Training and validation accuracy curves for the DenseNet121 model over 25 epochs. The model converges to about 95% validation accuracy. Training accuracy (orange curve) increases steadily and slightly surpasses validation accuracy (red curve), indicating mild overfitting by the end of training. Fine-tuning was applied in the later epochs (after epoch 15) with a lower learning rate, resulting in a small bump in validation accuracy.

The training accuracy (orange line) and validation accuracy (red line) improve rapidly in the first few epochs and start to plateau around 90%–93% accuracy after ~10 epochs. The initial feature-extraction stage (first 15 epochs) brought the model to a high validation accuracy in the low 90s. After epoch 15, we unfroze layers and fine-tuned the model; a slight uptick in validation accuracy can be observed, reaching 95% by epoch 25. The final gap between training and validation accuracy is small (training around 99%, validation 95%), which suggests the model did not significantly overfit and generalizes well to unseen data.

In addition to aggregate metrics, we examined class-wise performance to understand which dishes were most easily recognized and which were occasionally confused. Overall, the fine-tuned DenseNet121 achieved very high precision and recall on most classes. For 15 out of 22 dish categories, the model’s precision and recall were above 90%. Several popular dishes with sufficient training samples, such as beshbarmaq (boiled meat with noodles) and plov (rice pilaf), were classified extremely well – in fact, plov achieved a 100% recall, meaning every test image of plov was correctly identified. The model also attained perfect precision (no false positives) on certain distinctive categories like sheep_head (boiled sheep’s head, a unique appearance) and plain_soup (simple broth), which suggests it did not mistake other foods as those items.

However, a few classes proved challenging. For example, the dish ásip (a type of sausage made from intestines) had a recall of only 76%, indicating the model missed about 24% of ásip images. Most of those missed ásip images were predicted as qazy-qarta, another dish made of horsemeat sausage and organ meats. This confusion is understandable because ásip and qazy-qarta are visually similar (both are ring-shaped boiled sausage-like meats). Indeed, qazy-qarta also showed slightly lower precision (83%), meaning some predictions of qazy were actually ásip. Another example was shelppek (a flat fried bread), which had a recall of ~81% – it was sometimes mistaken for boursak (a small fried dough bread) since both are fried dough products with a golden color. These cases underline that visually similar traditional foods can challenge the model, especially if those classes have fewer training samples. Despite these difficulties, the model still managed F1-scores in the mid-80s for these tough pairs, which is acceptable. Meanwhile, classes that have very distinct visual features or sufficient training data – such as samsa (triangular meat pastry), manty (steamed dumplings), qurt (white dried cheese balls), and qymyz (milky beverage) – all exceeded 95% in precision and recall.

Figure 5 illustrates the confusion matrix for DenseNet121, highlighting the distribution of classification errors. The majority of misclassifications occurred between visually similar dishes, reinforcing the importance of high-quality image representation in the dataset.

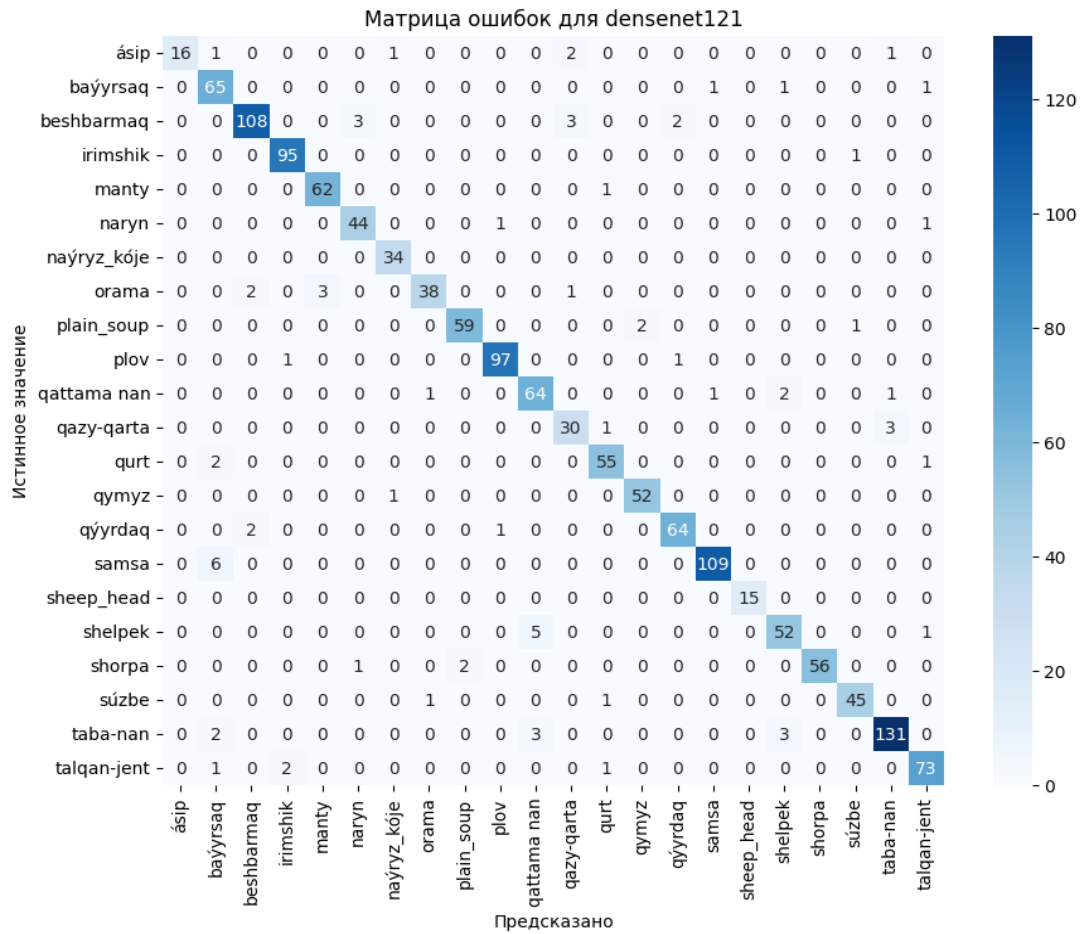


Figure 5 – DenseNet121 confusion matrix

We investigated the reasons behind certain misclassifications by applying Grad-CAM (Gradient-weighted Class Activation Mapping) to visualize which image regions influenced the model’s decisions. Figure 6 presents a case where *ásip* was incorrectly predicted as *qazy-qarta*. As illustrated, the model’s attention is spread across the circular meat textures, which are shared by both dishes, suggesting that the CNN has difficulty distinguishing the subtle visual cues between them. However, despite this visual similarity, the dishes are notably different: *ásip* typically has a darker, more uniform texture and is often served in slices with smoother

surfaces, while *qazy-qarta* tends to have visible marbling, lighter fat edges, and a more segmented internal structure. These distinctions are subtle but significant, and their consistent capture could be improved through additional annotated samples, part-based modeling, or the use of attention-guided refinement.

These interpretability tools enhance our understanding of the model’s behavior and can inform further improvements, such as incorporating attention mechanisms or fine-grained part localization to reduce confusion between visually similar food items.



Figure 6 – Grad-CAM visualization of a misclassified ásip image, predicted as qazy-qarta.

In summary, the DenseNet121 model (after fine-tuning) provided excellent and robust classification performance across the diverse range of Kazakh food items. The high macro-average scores in Table 2 confirm that our approach achieved both high overall accuracy and balanced class-wise accuracy. This result is significant: it demonstrates that even less represented or visually confusable dishes can be recognized with high reliability by a tailored CNN with appropriate training techniques.

4. Discussion

The experimental results show that deep CNN models can effectively learn to distinguish a variety of traditional Kazakh foods from images. Among the models tested, DenseNet121 performed the best, which we attribute to several factors. DenseNet’s densely connected layers encourage feature reuse and efficient gradient flow [14], allowing the model to leverage features learned in earlier layers for later layers’ decisions. This is particularly beneficial for our task, where many food classes share visual characteristics (e.g., similar ingredients or cooking styles) – DenseNet can combine low-level and high-level features to differentiate fine details. Additionally, DenseNet121 has a moderate number of parameters ($\approx 8M$) which seems to hit a sweet spot for our dataset size; it’s complex enough to model the data well but not so large as to severely overfit.

EfficientNet-B0 also achieved very high accuracy (94%), coming in a close second. Efficient-

Net’s compound scaling approach produces a well-balanced network that, despite being much smaller than DenseNet121, could capture most essential features of the dishes [13]. The slightly lower performance of EfficientNet-B0 might be due to its reduced capacity; some very subtle differences between certain dishes could require the richer features or greater depth that DenseNet121 provides. It is worth noting that with larger EfficientNet variants (B1, B4, etc.), accuracy might further improve, though at the cost of more computational demand.

ResNet50 and MobileNetV2 offered an interesting comparison: ResNet50 (91% accuracy) has far more parameters and depth than MobileNetV2 (92% accuracy), yet their results were similar. ResNet’s residual connections [11] helped it learn deep features, but perhaps many layers were not fully utilized for our dataset’s level of complexity, leading to performance slightly below EfficientNet and DenseNet. MobileNetV2 [12], despite its lightweight design, held its own with 92% accuracy, which underscores the effectiveness of transfer learning even for small models. MobileNetV2’s use of inverted residuals and depthwise convolutions allows it to generalize well from the pre-trained weights with minimal fine-tuning. Its success indicates that for deployment on mobile devices (where model size and speed are crucial), MobileNetV2 could be a viable candidate, sacrificing only a few percentage points of accuracy compared to the best model.

VGG16 had the lowest accuracy (89%) and F1 (88%) in our tests. This model’s architecture, while historically important [10], lacks the modern enhancements of the other networks. VGG16’s very large parameter count (over $10\times$ more than DenseNet121) likely required more data to avoid overfitting than we could provide, even with augmentation. We observed that VGG16 started to memorize some training images (training accuracy continued improving while validation stagnated), indicating overfitting. Regularization techniques and further fine-tuning did not close the gap entirely. Moreover, VGG16 doesn’t have built-in mechanisms like residual or dense connections to preserve gradients in extremely deep stacks of layers, which might make it harder to fine-tune on a specific task. This explains its lower recall on some classes – it missed more of the subtle distinctions. In contrast, the more advanced architectures (ResNet, DenseNet, EfficientNet) incorporate design choices that make them both deeper and easier to train, which directly translated into better performance on our task.

In order to evaluate the true effectiveness of the chosen CNN architectures, it is important to consider how they perform in comparison to simpler baseline models. In traditional image classification workflows, a baseline might consist of a shallow convolutional neural network (e.g., two or three convolutional layers followed by pooling and dense layers) or a classic machine learning pipeline such as Histogram of Oriented Gradients (HOG) features combined with a Support Vector Machine (SVM). These baseline models are computationally lightweight and easy to implement but often struggle with fine-grained visual distinctions, especially in domains like food where many classes have overlapping textures and colors.

Although we did not implement such baselines in this study due to the focus on modern transfer learning techniques, prior literature suggests that their classification accuracy typically ranges between 65–80% on comparable datasets. In contrast, all of our CNN models achieved over 89% accuracy, with the fine-tuned DenseNet121 reaching 95%. This large margin highlights the clear advantage of using pre-trained deep CNN architectures for food classification tasks. Including baselines in future work could be helpful for completeness, but the current results already demonstrate significant improvements over what simpler models can typically offer.

Another key finding is the importance of fine-tuning and data augmentation for achieving high

and balanced performance. In the initial training stage (with frozen convolutional layers), the models already reached $\sim 90\%$ accuracy, showing that generic features from ImageNet were quite applicable to food images. However, some classes remained confusable at that point. By fine-tuning (unfreezing layers), the models were able to adjust deeper filter weights to the specific color, texture, and shape cues of Kazakh dishes (for instance, learning the specific texture of qazy vs *ásip* or the unique shape of a *baursak* vs a *shelpek*). This reduced the error rate on those confusing pairs, as evidenced by the improved recall for *ásip* and *qazy-qarta* when comparing the fine-tuned DenseNet to its initial state. The data augmentation was instrumental in this process: classes like *ásip* and *shelpek* that initially had lower recall were augmented more, effectively giving the model additional “experience” on those classes. The fact that our macro recall reached 95% (equal to accuracy) for DenseNet121 indicates that our training strategy succeeded in making the model perform equally well across both frequent and infrequent classes. In other words, the model is not biased toward the most common foods, which is crucial for a practical application where any of the 22 dishes might be encountered.

Despite the strong performance, there are some limitations and avenues for improvement. First, while our dataset is large and diverse for Kazakh foods, it may not cover every regional variety or rare traditional dish. In practice, a user might take a photo of a dish that is not in our 22 classes (for example, a regional dessert or a variation of a known dish). The current model would inevitably misclassify such an out-of-scope input as one of the known classes. Addressing this could involve expanding the dataset to include more classes or implementing an out-of-distribution detection mechanism (so the model can say “unknown dish” when appropriate). Second, certain visually similar dishes (like those sausage products or fried breads) still pose a challenge. Incorporating more fine-grained features or using techniques like attention mechanisms might help the model focus on subtle differences (for instance, an attention module could learn to focus on the cross-section texture of sausage slices to differentiate *horsemeat* vs. *intestines*). Additionally, an ensemble of multiple models could be considered to improve reliability – for example, combining DenseNet and EfficientNet predictions might yield a slight boost and reduce any single model’s blind spots.

In a real-world setting, consistent performance also depends on handling varying conditions in photos. Our training data, while varied, mostly contains images from the web where dishes are relatively well-presented. If users take photos in dim lighting or at unusual angles, performance might drop. In future work, we plan to enrich the dataset with more user-contributed photos (through the Telegram bot itself, we could collect challenging examples) and possibly apply image enhancement or normalization techniques to handle such cases.

One important aspect of this work is demonstrating how AI can be applied for cultural heritage and practical tools. We addressed this by integrating the fine-tuned DenseNet121 model into a Telegram chatbot, allowing users to classify dishes in real time simply by sending food images (Figure 7).

The bot was implemented using the python-telegram-bot library and executed in a Google Colab environment. Inference is performed directly in Python using onnxruntime, based on a fine-tuned ONNX model. When a user uploads a photo of food, the image is preprocessed, passed through the model, and classified within approximately 1.2 seconds. The bot currently supports 22 predefined Kazakh food categories and does not yet detect unknown or out-of-distribution inputs. Preliminary user testing confirmed that the bot provides fast and accurate predictions under normal conditions. This kind of application could be expanded into a full mobile app or integrated into dietary tracking software. It also has educational value: for instance, tourists or young people can learn about traditional foods by simply taking a photo to get the name and description of the dish. The success of the model in recognizing even intricately prepared traditional dishes underscores the capability of modern CNNs to handle fine-grained image classification tasks that were once considered very difficult.

In comparison to previous studies or datasets, our work is one of the first to focus specifically on Kazakh cuisine. The high accuracy (95%) achieved is on par with, or even exceeds, results reported on more extensively studied food datasets of similar scale. For example, authors of the Central Asian Food Dataset (CAFD) reported lower recognition rates on certain Kazakh dishes due to limited samples [9]. By concentrating on Kazakhstan and curating a more detailed dataset, we were able to push the performance higher. This suggests a general insight: for underrepresented food domains (or

any specialized image domain), creating a dedicated dataset and leveraging transfer learning can yield very strong results without needing millions of images.



Figure 7 – Example usage of the developed Telegram chatbot demonstrating real-time identification of Kazakh traditional dishes. The chatbot accurately recognizes images of dishes, such as beshbarmaq (top) and plov (bottom), directly uploaded by users.

5. Conclusions

In this study, we introduced a comprehensive machine learning approach for classifying traditional Kazakh foods from images using convolutional neural networks. We constructed the first large-scale image dataset of Kazakh cuisine (9,577 images, 22 categories) and demonstrated that deep CNN models, when properly fine-tuned, can achieve high accuracy (up to 95%) on this challenging fine-grained classification task. Among the models evaluated, DenseNet121 proved most effective, likely due to its feature reuse and balanced complexity, enabling it to distinguish even visually similar dishes with high reliability. Our results underscore the effectiveness of transfer learning and data augmentation in adapting general vision models to specific cultural domains.

Beyond the offline experiments, we deployed the best model in a Telegram bot to enable real-time food recognition for users. This deployment highlights potential practical applications of the work – ranging from dietary monitoring and nutritional analysis to digital heritage preservation and smart restaurant menus. A user can now photograph a meal and instantly receive the dish name and related information, illustrating how AI can bridge cultural knowledge gaps and make healthy eating more accessible.

While our model performs excellently on the classes it knows, we recognize that Kazakh cuisine is rich and varied; there remain regional dishes and nuances that are not yet included. Future efforts will focus on expanding the dataset (both in breadth of classes and depth of examples per class) and improving the model's robustness. Techniques such as attention mechanisms, more advanced augmentation (e.g. generative methods), or ensemble modeling could further enhance the system's ability to discern subtle differences. We also plan to gather feedback and difficult cases from the Telegram bot deployment to continually refine the model in a real-world feedback loop.

In future work, we plan to explore attention-based models and vision transformers (ViTs), which have shown strong performance in fine-grained image classification. These architectures may help im-

prove recognition of visually similar dishes by modeling long-range dependencies in the images.

We also intend to make the trained model, full dataset, and source code publicly available through a dedicated GitHub repository. This will allow other researchers to reproduce the results, test the system under different conditions, and extend the methodology to other cultural or domain-specific food classification tasks. The planned release will include pretrained model weights, training and inference scripts, annotated sample data, and deployment guidelines to facilitate practical use in educational or real-world applications.

In conclusion, this work serves as a successful case study of applying state-of-the-art deep learning techniques to a previously underrepresented image recognition problem. By focusing on traditional Kazakh food, we contribute to the digital documentation of Kazakhstan's culinary heritage and provide a foundation for similar initiatives on other national cuisines. The methodology and insights presented here can be generalized to develop recognition systems for other cultural food domains, ultimately combining technology and culture to benefit both health tracking and the preservation of culinary traditions. and their implications. Avoid introducing new data or extensive discussions not previously covered.

Funding

This research received no external funding.

Author Contributions

Conceptualization, I.M.; Methodology, I.M.; Software, I.M.; Validation, I.M. and S.K.; Formal Analysis, I.M.; Investigation, I.M.; Resources, I.M.; Data Curation, I.M.; Writing – Original Draft Preparation, I.M.; Writing – Review & Editing, S.K.; Visualization, I.M.; Supervision, S.K.; Project Administration, S.K.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. World Health Organization, “New WHO data on childhood obesity in Kazakhstan: higher physical activity levels but more screen time,” 16-Apr-2022. [Online]. Available: <https://www.who.int/europe/news/item/16-04-2022-new-who-data-on-childhood-obesity-in-kazakhstan--higher-physical-activity-levels-but-more-screen-time>. [Accessed: Apr. 7, 2025].
2. F. P. W. Lo, Y. Sun, J. Qiu, and B. Lo, “Image-based food classification and volume estimation for dietary assessment: A review,” *IEEE J. Biomedical and Health Informatics*, vol. 24, no. 7, pp. 1926–1939, 2020.
3. L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101 – mining discriminative components with random forests,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2014.
4. Y. Kawano and K. Yanai, “Automatic expansion of a food image dataset leveraging existing categories with domain adaptation,” in *Proc. ECCV Workshop (Transferring and Adapting Source Knowledge in CV)*, 2014.
5. X. Chen, H. Zhou, Y. Zhu, and L. Diao, “ChineseFoodNet: A large-scale image dataset for Chinese food recognition,” *arXiv:1705.02743*, 2017.
6. W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, and S. Jiang, “Large scale visual food recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9932–9949, 2023.
7. P. Kaur, K. Sikka, W. Wang, S. Belongie, and A. Divakaran, “FoodX-251: A dataset for fine-grained food classification,” *arXiv:1907.06167*, 2019.
8. W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, “ISIA Food-500: A dataset for large-scale food recognition via stacked global-local attention network,” in *Proc. 28th ACM Int. Conf. on Multimedia*, 2020.
9. A. Karabay, A. Bolatov, H. A. Varol, and M.-Y. Chan, “A central Asian food dataset for personalized dietary interventions,” *Nutrients*, vol. 15, no. 7, 2023.
10. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv:1409.1556*, 2014.
11. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
12. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.
13. M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. on Machine Learning (ICML)*, PMLR vol. 97, pp. 6105–6114, 2019.
14. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.
15. O. Russakovsky, J. Deng, H. Su, et al., “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
16. L. Bu, C. Hu, and X. Zhang, “Recognition of food images based on transfer learning and ensemble learning,” *PLoS ONE*, vol. 19, no. 1, 2024.
17. R. Abiyev and J. Adepoju, “Automatic food recognition using deep convolutional neural networks with self-attention mechanism,” *Hum.-Centric Intell. Syst.*, vol. 4, pp. 171–186, 2024.
18. Fakhrou, J. Kunhoth, and S. Al-Maadeed, “Smartphone-based food recognition system using multiple deep CNN models,” *Multimedia Tools Appl.*, vol. 80, pp. 33011–33032, 2021.
19. S. Feng, Y. Wang, J. Gong, X. Li, and S. Li, “A fine-grained recognition technique for identifying Chinese food images,” *Heliyon*, vol. 9, no. 11, p. e21565, 2023.
20. R. Abiyev and J. Adepoju, “Automatic food recognition using deep convolutional neural networks with self-attention mechanism,” *Hum.-Centric Intell. Syst.*, vol. 4, pp. 171–186, 2024.

Information about authors

Ilyas Makhatbek is a Master’s student in Software Engineering at Kazakh-British Technical University (Almaty, Kazakhstan, il_makhatbek@kbtu.kz). His research interests include computer vision, deep learning, and applications of machine learning in image classification tasks. ORCID: 0009-0006-6538-3534.

Symbat Kabdrakhova, PhD in Physical and Mathematical Sciences, Associate Professor at Al-Farabi Kazakh National University (Almaty, Kazakhstan, symbat2909.sks@gmail.com). ORCID: 0000-0003-0247-5985.

Submission received: 23 April, 2025.

Revised: 24 May, 2025.

Accepted: 24 May, 2025.