

SCO INTERNATIONAL AI OLYMPIAD

QUESTION PAPER - SET S

2025-26 | Answer Key and Detailed Explanations Included

A professionally formatted AI Olympiad practice paper for senior school learners.

- Deep learning, CNNs, RNNs, robotics, IoT, responsible AI, OpenCV, Keras, and Python reasoning
- Clean question blocks with code snippets, options, answer keys, and explanations
- Designed for website PDF viewing, downloads, teacher review, and student preparation

Deep AI	Robotics	IoT	Ethics	OpenCV
Keras	Python	Data	Sustainability	ML

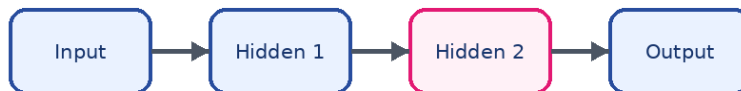
General AI Foundations

Deep learning basics, CNN/RNN differences, IoT optimization, OpenCV tracking, healthcare chatbot ethics, and core model interpretation.

Q1

In a Deep Neural Network (DNN), vanishing gradients are most likely to occur when:

Deep neural network: inputs, hidden layers, output



Backpropagation adjusts weights to reduce prediction error.

- A. The activation function is ReLU and the learning rate is very high.
- B. The activation function is sigmoid, and the weights are initialized randomly.
- C. There are fewer hidden layers, and dropout is used.
- D. The optimizer is Adam and the network is shallow.

Answer: (b) The activation function is sigmoid, and the weights are initialized randomly.

Explanation: Vanishing gradients often occur in deep networks with sigmoid or tanh activation functions, where gradients shrink exponentially in backpropagation. This issue can be mitigated using ReLU or weight initialization techniques.

Q2

What is the key difference between CNNs and RNNs?

- A. CNNs are used for time-series data, and RNNs are used for image data.
- B. CNNs use convolutional filters, while RNNs use memory cells for sequential data.
- C. CNNs perform poorly on image data compared to RNNs.
- D. RNNs cannot be used for video processing tasks.

Answer: (b) CNNs use convolutional filters, while RNNs use memory cells for sequential data.

Explanation: CNNs are effective for spatial data like images due to convolutional filters, while RNNs process sequential data like time-series or natural language due to memory cells that store contextual information.

Q3

In an IoT-based robotics application, AI can be integrated to optimize energy usage by:

- A. Increasing the number of connected devices.
- B. Predicting device usage patterns using machine learning models.
- C. Adding redundant sensors to the system.
- D. Increasing the complexity of neural networks used in the robot.

Answer: (b) Predicting device usage patterns using machine learning models.

Explanation: AI can optimize energy usage in IoT systems by analyzing usage patterns and predicting future needs, reducing unnecessary power consumption.

Q4

In OpenCV, which method is best suited for real-time object tracking in video streams?

- A. Hough Transform.
- B. Optical Flow.
- C. Histogram Equalization.
- D. Canny Edge Detection.

Answer: (b) Optical Flow.

Explanation: Optical Flow tracks the motion of objects between consecutive frames in video, making it suitable for real-time applications like autonomous navigation.

Q5

Which of these represents a critical ethical concern in deploying AI-based chatbots for healthcare?

- A. Difficulty in integrating with IoT devices.
- B. Over-reliance on labeled datasets.
- C. Potential for bias in diagnosis recommendations due to training data.
- D. Low computational efficiency compared to traditional models.

Answer: (c) Potential for bias in diagnosis recommendations due to training data.

Explanation: AI chatbots in healthcare risk being biased if the training data reflects inaccuracies or demographic biases, which can lead to unreliable recommendations.

Q6

For autonomous navigation, an AI robot uses LIDAR and a CNN-based vision system. What is the likely role of the CNN?

- A. To detect obstacles and classify objects in the environment.
- B. To control the movement of the robot's wheels.
- C. To calculate the shortest path between two points.
- D. To estimate the battery usage of the robot.

Answer: (a) To detect obstacles and classify objects in the environment.

Explanation: CNNs excel at image classification and object detection, making them suitable for identifying obstacles and features in a robot's environment.

Q7

While training an AI model, what does the following Python code snippet represent?

```
from keras.layers import Dense
from keras.models import Sequential
```

```
model = Sequential([
```

```
Dense(128, activation='relu', input_shape=(784,)), Dense(64, activation='relu'), Dense(10, activation='softmax')
```

```
])
```

- A. A CNN model with convolutional and pooling layers.
- B. An RNN model for sequential data.
- C. A feedforward neural network for multi-class classification.
- D. An unsupervised clustering model.

Answer: (c) A feedforward neural network for multi-class classification.

Explanation: This code defines a sequential feedforward neural network with two hidden layers and a softmax output layer for multi-class classification.

Q8

In a Capstone Project on AI in Healthcare, an object detection model flagged 95% of cancerous tumors but incorrectly identified 20% of benign growths as cancerous. This issue is an example of:

- A. High precision and low recall.
- B. High recall and low precision.
- C. Overfitting.
- D. Underfitting.

Answer: (b) High recall and low precision.

Explanation: High recall indicates the model detects most positive cases but low precision shows a high number of false positives.

Q9

Which technique would you apply to mitigate overfitting in a deep learning model?

- A. Reduce the dataset size.
- B. Use a higher learning rate.
- C. Implement dropout layers and regularization.
- D. Remove layers from the network.

Answer: (c) Implement dropout layers and regularization.

Explanation: Dropout and regularization techniques like L2 prevent overfitting by penalizing large weights and reducing reliance on specific neurons.

Q10

Which of these tasks is least suitable for a standard NLP Transformer model without vision-specific adaptation?

- A. Text generation in natural language processing.
- B. Sequence-to-sequence translation tasks.
- C. Image recognition in unstructured image datasets.
- D. Summarization of lengthy articles.

Answer: (c) Image recognition in unstructured image datasets.

Explanation: Standard text Transformer models are strongest for language tasks such as generation, translation, and summarization. Image recognition generally requires computer-vision architectures such as CNNs or vision-adapted Transformers. Therefore, among these choices, unstructured image recognition is the least suitable for a standard NLP Transformer.

Case Studies: Applied AI Systems

Autonomous vehicles, healthcare diagnosis, recommender systems, mobile translation, smart surveillance, smart homes, drones, predictive maintenance, robotics, and autonomous ethics.

Q11

Case Study 1: Autonomous Vehicle Navigation The autonomous vehicle system uses CNNs to process image data of traffic signs and RNNs to predict next actions. A resource-limited scenario requires using a lightweight model. Which architecture modification would best balance accuracy and computational efficiency?

- A. Replace the CNN with a single-layer perceptron to reduce computation.
- B. Use a MobileNet-based CNN and a Gated Recurrent Unit (GRU) instead of a traditional RNN.
- C. Increase the depth of the CNN while using a standard RNN.
- D. Train the system with reduced image resolution while maintaining the current architecture.

Answer: (b) Use a MobileNet-based CNN and a Gated Recurrent Unit (GRU) instead of a traditional RNN.

Explanation: MobileNet is optimized for computational efficiency, and GRUs simplify RNN architecture without significantly compromising performance.

Q12

Case Study 2: Healthcare Diagnosis Using Neural Networks

If the neural network misclassifies a rare disease due to insufficient data in the training set, which approach should be adopted to improve accuracy without redesigning the full model architecture?

- A. Introduce synthetic data for rare cases using generative adversarial networks (GANs).
- B. Replace the CNN with an ensemble of simpler classifiers.
- C. Decrease the number of convolutional layers to avoid overfitting.
- D. Adjust the learning rate dynamically during training.

Answer: (a) Introduce synthetic data for rare cases using generative adversarial networks (GANs).

Explanation: Synthetic examples for rare cases can expand the limited class representation and support targeted fine-tuning without redesigning the complete model architecture. This helps the model learn features of underrepresented disease cases more effectively.

Q13

Case Study 3: E-Commerce Recommendation Systems

The recommendation system detects inconsistent purchase patterns due to noise in the browsing history data. What pre-processing step would most effectively reduce noise and improve RNN predictions?

- A. Use Principal Component Analysis (PCA) on the browsing sequence.
- B. Apply a moving average filter to smooth the sequence data.
- C. Implement dropout layers in the RNN during inference.
- D. Remove all data points that deviate significantly from the mean.

Answer: (b) Apply a moving average filter to smooth the sequence data.

Explanation: Smoothing data with a moving average reduces noise while retaining the overall trend, improving the RNN's ability to make accurate predictions.

Q14

Case Study 4: Real-Time Language Translation For a resource-constrained mobile device, the Transformer-based translation system struggles with memory efficiency. Which modification would most effectively balance memory usage and translation quality?

- A. Use a smaller vocabulary size in the encoder and decoder.
- B. Replace self-attention layers with convolutional layers.
- C. Use a quantized version of the Transformer model.
- D. Limit the sequence length of the input text to the first 10 words.

Answer: (c) Use a quantized version of the Transformer model.

Explanation: Model quantization reduces the memory footprint of a Transformer without significant loss in translation quality.

Q15

Case Study 5: Smart Surveillance Systems The surveillance system must detect suspicious behavior while adhering to privacy regulations. Which approach balances accuracy and privacy concerns?

- A. Use federated learning to train the CNN model across distributed datasets.
- B. Store all video data centrally to improve model performance.
- C. Replace the CNN with a single-layer network for faster analysis.
- D. Use a pre-trained CNN without additional training on the dataset.

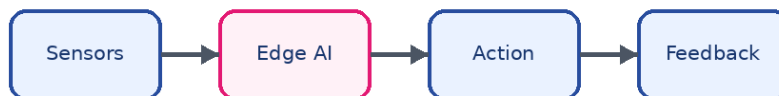
Answer: (a) Use federated learning to train the CNN model across distributed datasets.

Explanation: Federated learning allows model training on decentralized data, ensuring privacy while improving the model's ability to detect suspicious activity.

Q16

Optimizing Energy Efficiency in Smart Homes A smart home system uses IoT sensors and AI algorithms to optimize energy usage. It monitors temperature, occupancy, and appliance usage. To minimize energy consumption while ensuring comfort, the system must decide when to turn on/off appliances. Which approach would balance these constraints most effectively?

AI + IoT + robotics: adaptive decision loop



Low-latency sensing and learning support safe real-time responses.

- A. Use a rule-based system with fixed thresholds for temperature and occupancy.
- B. Apply reinforcement learning where the AI learns optimal actions based on energy usage feedback.
- C. Implement a k-means clustering algorithm to group similar appliance usage patterns.
- D. Use a decision tree trained on historical energy consumption data.

Answer: (b) Apply reinforcement learning where the AI learns optimal actions based on energy usage feedback.

Explanation: Reinforcement learning enables the AI to dynamically learn the best strategies for balancing energy efficiency and comfort by optimizing rewards based on real-time feedback. Rule-based systems and static models cannot adapt to varying scenarios effectively.

Q17

Autonomous Drone Navigation in Dynamic Environments An autonomous drone uses AI for navigation in a warehouse to deliver packages. The drone employs LIDAR sensors for obstacle detection and neural networks for path planning. During high congestion, its pathfinding slows down. Which enhancement would best address this issue?

- A. Replace LIDAR with a high-resolution camera to improve object detection.
- B. Use a convolutional neural network (CNN) combined with a recurrent neural network (RNN) to predict dynamic obstacles.
- C. Add more LIDAR sensors to reduce blind spots.
- D. Pre-program the drone with fixed paths to avoid real-time computation.

Answer: (b) Use a convolutional neural network (CNN) combined with a recurrent neural network (RNN) to predict dynamic obstacles.

Explanation: CNNs excel at processing visual or spatial data, and RNNs can model sequential information to predict the movement of dynamic obstacles. This combination allows the drone to plan paths more efficiently in congested environments.

Q18

IoT-Based Predictive Maintenance in Manufacturing In a smart factory, IoT sensors monitor equipment for signs of wear and tear. An AI model predicts machine failures using the data. Which scenario presents the greatest challenge for the AI system, and what solution would mitigate it?

- A. Imbalanced data with few failure events; use data augmentation or synthetic data generation.
- B. Sensors reporting conflicting data; increase the frequency of sensor calibration.
- C. High dimensionality of sensor data; apply feature selection techniques to reduce complexity.

D. Sudden equipment failures without warning signs; use transfer learning with pre-trained models.

Answer: (a) Imbalanced data with few failure events; use data augmentation or synthetic data generation.

Explanation: Imbalanced data is common in predictive maintenance, as failures are rare compared to normal operation. Data augmentation or generating synthetic failure cases can provide the model with enough examples to improve prediction accuracy.

Q19

Multi-Agent Robotic System for Task Allocation A fleet of warehouse robots collaborates to transport goods efficiently. AI assigns tasks based on proximity and workload. During peak hours, task allocation becomes suboptimal, leading to delays. What enhancement would make the system more efficient?

- A. Implement a genetic algorithm to optimize task allocation in real time.
- B. Use a heuristic-based approach to assign tasks sequentially.
- C. Train a deep Q-network (DQN) for each robot to make individual decisions.
- D. Prioritize robots based on their battery levels and ignore proximity.

Answer: (a) Implement a genetic algorithm to optimize task allocation in real time.

Explanation: Genetic algorithms can efficiently explore large search spaces for optimal task assignments, even in dynamic environments, ensuring that robots collaborate effectively under high workload conditions.

Q20

Ethical Challenges in Autonomous Vehicles An autonomous car uses AI and IoT sensors to navigate urban environments. It encounters a dilemma where it must choose between two undesirable outcomes: hitting a pedestrian or colliding with another vehicle. Which ethical framework would best guide its decision-making algorithm?

- A. Utilitarianism: Minimize overall harm by selecting the option with the least casualties.
- B. Deontological ethics: Follow a strict rule, such as never hitting pedestrians.
- C. Random decision-making to avoid bias in prioritization.

D. Allow the driver to intervene and take control in such scenarios.

Answer: (a) Utilitarianism: Minimize overall harm by selecting the option with the least casualties.

Explanation: Utilitarianism provides a consistent ethical framework for minimizing harm, which is critical in programming autonomous systems. Random decisions or rigid rules can lead to inconsistent or ethically questionable outcomes.

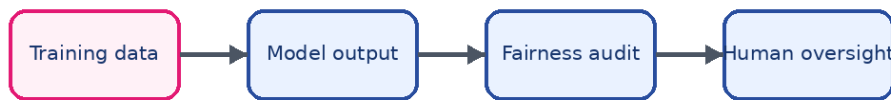
Responsible AI, Sustainability, and Real-World Deployment

Fairness, sustainability, privacy, healthcare equity, low-light object detection, contextual chatbots, medical deployment gaps, wildlife conservation, and smart security.

Q21

Biased Hiring Algorithm in a Recruitment System A company uses an AI-based recruitment tool to screen resumes. Over time, it is discovered that the algorithm disproportionately rejects applications from certain demographic groups due to biased historical data. How should the company address this issue?

Responsible AI: detect, measure, and mitigate bias



Representative data, transparency, privacy, and review reduce unfair outcomes.

- A. Increase the training dataset size to improve generalization.
- B. Implement fairness-aware algorithms and retrain the model using unbiased data.
- C. Remove demographic information from the resumes before feeding them into the AI.
- D. Adjust the decision thresholds to favor underrepresented groups.

Answer: (b) Implement fairness-aware algorithms and retrain the model using unbiased data.

Explanation: Merely increasing dataset size or removing demographic information does not address systemic biases. Fairness-aware algorithms analyze and mitigate bias, while retraining with an unbiased dataset ensures equitable decision-making.

Q22

Sustainability of Large AI Models Training a large AI model consumes significant energy, with some models producing carbon footprints equivalent to dozens of cars over a year. Which solution most effectively balances AI development and sustainability?

- A. Use smaller models with lower computational requirements, even if accuracy decreases slightly.
- B. Transition training processes to energy-efficient GPUs and renewable energy sources.
- C. Limit AI model training to institutions with carbon offset programs.
- D. Restrict the use of large AI models to critical applications only.

Answer: (b) Transition training processes to energy-efficient GPUs and renewable energy sources.

Explanation: Transitioning to energy-efficient hardware and renewable energy reduces environmental impact without sacrificing AI innovation. While smaller models help, they may not meet performance demands for critical tasks.

Q23

Data Privacy in AI-Powered Smart Devices A smart voice assistant records conversations to improve its AI responses. Users later discover that some recordings were accessed by developers for quality checks without explicit consent. What is the most ethical way to handle such situations?

- A. Encrypt all recordings and store them for developer access when needed.
- B. Notify users of the potential for manual reviews during setup and provide an opt-out option.
- C. Anonymize all recordings to protect user identity before analysis.
- D. Stop manual reviews entirely to maintain privacy.

Answer: (b) Notify users of the potential for manual reviews during setup and provide an opt-out option.

Explanation: Transparency and user consent are core to ethical AI. While anonymization helps, it may not fully address privacy concerns without user awareness and control.

Q24

Autonomous Vehicles and Moral Dilemmas An autonomous vehicle must decide whether to swerve to avoid hitting a pedestrian but risks endangering its passenger. If the AI system is programmed to prioritize passenger safety, what ethical issue arises?

- A. The AI violates the principle of minimizing harm.
- B. The AI risks legal challenges by prioritizing passengers over pedestrians.
- C. The AI violates transparency by not disclosing its prioritization logic.
- D. The AI introduces bias by favoring paying users (passengers) over non-paying ones (pedestrians).

Answer: (a) The AI violates the principle of minimizing harm.

Explanation: Ethical AI prioritizes minimizing harm for all parties involved, not just a specific group. Programming bias to favor passengers disregards broader societal ethics.

Q25

Algorithmic Decision-Making in Healthcare An AI system predicts the urgency of patients in an emergency room. The model underestimates the severity for underrepresented groups due to data imbalance. Which approach best ensures fairness and accuracy?

- A. Collect more data from underrepresented groups to improve the model's training.
- B. Use separate models for different demographic groups to ensure tailored predictions.
- C. Incorporate fairness metrics into the evaluation criteria and adjust predictions accordingly.
- D. Perform manual reviews for underrepresented groups alongside AI predictions.

Answer: (c) Incorporate fairness metrics into the evaluation criteria and adjust predictions accordingly.

Explanation: Fairness metrics identify and mitigate bias in predictions, ensuring equitable treatment for all groups. Separate models may inadvertently introduce further bias, while manual reviews are not scalable.

Q26

Real-Time Object Detection in Traffic Management A city uses an AI-powered object detection system to monitor traffic violations like jumping signals and speeding. During testing, it's observed that the system struggles to detect vehicles in low-light conditions. Which approach is most effective to enhance its performance?

- A. Train the model on a larger dataset containing low-light scenarios.
- B. Apply histogram equalization to all images to improve brightness and contrast.
- C. Use a pre-trained model optimized for night-time object detection.
- D. Integrate thermal imaging cameras for better detection in low-light conditions.

Answer: (d) Integrate thermal imaging cameras for better detection in low-light conditions.

Explanation: While training on low-light scenarios or using pre-trained models can help, thermal imaging directly addresses the problem by detecting heat signatures, independent of lighting conditions.

Q27

AI Chatbots in Customer Support A company deploys an AI chatbot to handle customer queries. Over time, customers complain about the chatbot providing incorrect or irrelevant responses. Analysis reveals that the chatbot frequently misunderstands context. How can this issue be resolved effectively?

- A. Increase the size of the training dataset with diverse conversational examples.
- B. Transition to a transformer-based architecture like GPT for better contextual understanding.
- C. Add pre-defined responses for commonly misunderstood queries.
- D. Enable human intervention for queries the chatbot cannot resolve.

Answer: (b) Transition to a transformer-based architecture like GPT for better contextual understanding.

Explanation: Transformer-based architectures excel in understanding context and nuances, making them ideal for improving chatbot accuracy. Increasing dataset size or enabling human intervention may not fully address the contextual issues.

Q28

AI in Healthcare for Disease Diagnosis An AI system is developed to detect early signs of lung cancer from chest X-rays. The model performs well on the test data but fails when deployed in hospitals due to differences in image quality from older machines. What is the best solution to address this problem?

- A. Use transfer learning with a diverse dataset including low-quality images.
- B. Apply image preprocessing techniques like noise reduction and normalization.
- C. Upgrade hospital imaging machines to generate better-quality X-rays.
- D. Use a different AI model specifically trained on older imaging data.

Answer: (a) Use transfer learning with a diverse dataset including low-quality images.

Explanation: Transfer learning allows the model to adapt to new domains by leveraging knowledge from a pre-trained model, ensuring it performs well across varying image qualities.

Q29

AI for Wildlife Conservation An AI model uses drone footage to count and classify endangered species in a forest. During deployment, the system fails to differentiate between similar-looking species. Which approach will most effectively improve classification accuracy?

- A. Train a new model with a focus on the similar-looking species.
- B. Use a hybrid model combining CNNs and RNNs for spatial and temporal data analysis.
- C. Increase the resolution of drone cameras to capture more detailed images.
- D. Add manual classification for species the model cannot confidently identify.

Answer: (b) Use a hybrid model combining CNNs and RNNs for spatial and temporal data analysis.

Explanation: A hybrid model can analyze spatial features (appearance) and temporal patterns (movement), making it better suited for differentiating similar-looking species in dynamic environments.

Q30

AI-Powered Smart Home Security System A smart home security system uses AI to detect unusual activities and alert homeowners. It frequently raises false alarms for activities like pets moving or trees swaying. Which solution is most effective to reduce false positives?

- A. Retrain the model to exclude pet and tree movement from unusual activities.
- B. Introduce a second AI layer to validate detections before sending alerts.
- C. Use motion sensors instead of cameras to focus on significant movements.
- D. Allow homeowners to label false positives for incremental model improvement.

Answer: (b) Introduce a second AI layer to validate detections before sending alerts.

Explanation: A secondary validation layer refines the system's decision-making by adding another filter, ensuring only genuine alerts are sent. Retraining or user labeling may take longer to show results.

Computer Vision, CNNs, and Keras Practice

Image preprocessing, CNN initialization, OpenCV edge detection, softmax activation, data augmentation, batch normalization, IoU, feature matching, GRUs, and transfer learning.

Q31

Image Preprocessing in OpenCV You are preprocessing images for a CNN model using OpenCV. To normalize the pixel values to a range between 0 and 1, which of the following is the most efficient approach?

- A. `image = cv2.normalize(image, None, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX)`
- B. `image = image / 255.0`
- C. `image = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)[1]`
- D. `image = cv2.equalizeHist(image)`

Answer: (b) image = image / 255.0

Explanation: Dividing by 255.0 normalizes pixel values to the range [0, 1], which is standard for CNNs. The other methods are unrelated to normalization for this purpose.

Q32

You are designing a convolutional neural network for image classification. What happens if all the filters in a convolutional layer are initialized with the same weights?

- A. The network converges faster.
- B. All filters learn the same features, reducing the network's effectiveness.
- C. The model overfits to the training data.
- D. The filters cancel each other out, and the layer outputs zero.

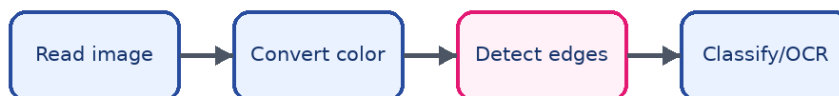
Answer: (b) All filters learn the same features, reducing the network's effectiveness.

Explanation: When filters are initialized identically, the gradients for all filters are the same, leading them to learn identical features, defeating the purpose of using multiple filters.

Q33

OpenCV for Edge Detection Which of the following is the most effective way to perform edge detection using OpenCV for preparing input data for object detection?

Computer vision preprocessing pipeline



Clean image inputs improve reliable object detection and recognition.

- A. `cv2.Canny(image, 50, 150)`
- B. `cv2.blur(image, (5, 5))`
- C. `cv2.dilate(image, kernel, iterations=1)`

D. `cv2.equalizeHist(image)`

Answer: (a) `cv2.Canny(image, 50, 150)`

Explanation: Canny edge detection is a robust algorithm specifically designed for detecting edges, making it the best choice for this task.

Q34

CNN Activation Functions You are training a CNN for multi-class image classification. Which activation function is most appropriate for the output layer?

A. ReLU

B. Sigmoid

C. Softmax

D. Tanh

Answer: (c) Softmax

Explanation: Softmax is used in the output layer of a multi-class classification model as it converts logits into probabilities, summing up to 1.

Q35

Data Augmentation in Keras Which Python code snippet correctly performs data augmentation for a dataset using Keras?

A. `datagen = ImageDataGenerator(rotation_range=40, width_shift_range=0.2, height_shift_range=0.2)`

```
augmented_images = datagen.flow(images, batch_size=32)
```

B. `datagen = ImageDataGenerator()`

```
augmented_images = datagen(images)
```

C. `datagen = ImageDataGenerator(horizontal_flip=True)`

```
augmented_images = datagen.fit(images)
```

D. `datagen = ImageDataGenerator(rescale=1/255.0)`

```
augmented_images = datagen.flow(images, labels, batch_size=32)
```

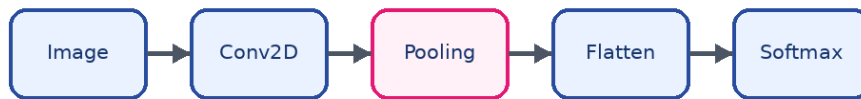
Answer: (a)

Explanation: Option (a) uses ImageDataGenerator with rotation and shift parameters and then creates batches through flow(). These settings generate transformed training examples, which is the purpose of data augmentation.

Q36

Batch Normalization in CNNs What is the primary purpose of batch normalization in CNNs?

CNN image classification: features to probability



Pooling reduces spatial dimensions; softmax gives class probabilities.

A. Prevent overfitting by reducing the model capacity.

B. Standardize the inputs to a layer to speed up convergence.

C. Reduce the size of the feature maps.

D. Increase the depth of the network without increasing computational complexity.

Answer: (b) Standardize the inputs to a layer to speed up convergence.

Explanation: Batch normalization normalizes the inputs for each layer, stabilizing learning and improving convergence speed.

Q37

IoU Metric in Object Detection What does the Intersection over Union (IoU) metric represent in object detection tasks?

- A. The percentage of correctly classified objects.
- B. The overlap between predicted and ground truth bounding boxes relative to their union.
- C. The ratio of true positives to false positives.
- D. The area under the ROC curve.

Answer: (b) The overlap between predicted and ground truth bounding boxes relative to their union.

Explanation: IoU measures how well the predicted bounding box aligns with the ground truth, a key metric in object detection.

Q38

OpenCV for Feature Matching Which method in OpenCV is used for feature matching using SIFT descriptors?

- A. `cv2.matchTemplate()`
- B. `cv2.BFMatcher().match()`
- C. `cv2.Canny()`
- D. `cv2.findContours()`

Answer: (b) `cv2.BFMatcher().match()`

Explanation: BFMatcher (Brute Force Matcher) is used for matching SIFT descriptors, crucial for tasks like object recognition.

Q39

RNN for Time Series Prediction An RNN model is trained to predict stock prices. The model struggles with vanishing gradients during backpropagation. Which architecture is most appropriate to resolve this issue?

- A. Simple RNN
- B. GRU
- C. Feedforward Network
- D. CNN

Answer: (b) GRU

Explanation: GRU (Gated Recurrent Unit) effectively mitigates the vanishing gradient problem by using gating mechanisms, making it suitable for sequential data tasks like stock prediction.

Q40

Transfer Learning in CNNs Which Python code snippet correctly implements transfer learning with a pre-trained ResNet50 model for image classification?

A. `from tensorflow.keras.applications import ResNet50`

```
model = ResNet50(weights='imagenet', include_top=True)
model.add(Dense(10, activation='softmax'))
```

B. `from tensorflow.keras.applications import ResNet50`

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense

base_model = ResNet50(weights='imagenet', include_top=False)
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dense(10, activation='softmax')
])
```

C. `from tensorflow.keras.applications import ResNet50`

```
model = ResNet50(weights='imagenet', include_top=False)
model.compile(optimizer='adam', loss='categorical_crossentropy')
```

D. `from tensorflow.keras.applications import ResNet50`

```
model = ResNet50(weights=None, include_top=False)
model.add(Dense(10, activation='softmax'))
```

Answer: (b)

Explanation: Option (b) correctly loads the ImageNet-pretrained ResNet50 base without its original classification head, adds a pooling layer to convert feature maps into a vector, and then adds a new softmax classifier for the target classes.

Python Programming and Data Science Reasoning

List comprehensions, mutable defaults, decorators, slicing, generators, lambda/map/filter, dictionary sorting, dictionary comprehensions, zip, and permutations.

Q41

Comprehension with Conditionals What will be the output of the following code?

```
matrix = [[x for x in range(5) if x % 2 == 0] for _ in range(3)]
print(matrix)
```

A. `[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]`

B. `[[0, 2, 4], [0, 2, 4], [0, 2, 4]]`

C. `[[1, 3], [1, 3], [1, 3]]`

D. `[[0, 4], [0, 4], [0, 4]]`

Answer: (b) `[[0, 2, 4], [0, 2, 4], [0, 2, 4]]`

Explanation: The inner list comprehension generates numbers divisible by 2, and the outer list repeats this list 3 times.

Q42

Function Argument Evaluation What will be the output of the following code?

```
def func(a, b=[]):
```

```
    b.append(a)
```

```

return b

print(func(1))
print(func(2))

```

- A. [1] and [2]
- B. [1] and [1, 2]
- C. [1] and [2] (separate lists)
- D. [1] and [1, 1]

Answer: (b) [1] and [1, 2]

Explanation: Default mutable arguments (like lists) retain changes across function calls. b is shared between calls.

Q43

Decorator Behavior What does the following decorator function do?

```

def log_decorator(func):
    def wrapper(*args, **kwargs):
        print(f"Calling {func.__name__}")
        return func(*args, **kwargs)
    return wrapper

@log_decorator
def add(a, b):
    return a + b

print(add(3, 4))

```

- A. Outputs Calling add followed by 7
- B. Outputs add followed by Calling add
- C. Outputs 7 followed by Calling add
- D. Raises an error

Answer: (a) Outputs Calling add followed by 7

Explanation: The decorator wraps the add function, logging its call before execution.

Q44

Advanced List Slicing What will be the output of the following code?

```
nums = [1, 2, 3, 4, 5, 6]
print(nums[::-2])
```

- A. [6, 4, 2]
- B. [6, 5, 4, 3, 2, 1]
- C. [5, 3, 1]
- D. [6, 4, 2, 0]

Answer: (a) [6, 4, 2]

Explanation: `::-2` reverses the list and skips every other element.

Q45

Generator Expressions What does the following code output?

```
gen = (x**2 for x in range(4))
print(next(gen), next(gen))
```

- A. 0 and 1
- B. 1 and 4
- C. 0 and 4
- D. 1 and 9

Answer: (a) 0 and 1

Explanation: The generator produces squares of numbers starting from 0. The first two elements are 0^2 and 1^2 .

Q46

Lambda with Map and Filter What will be the output of the following code?

```
nums = [1, 2, 3, 4]
result = list(map(lambda x: x**2, filter(lambda x: x % 2 == 0, nums)))
print(result)
```

A. [1, 4, 9, 16]

B. [4, 16]

C. [2, 4]

D. [1, 9]

Answer: (b) [4, 16]

Explanation: filter selects even numbers [2, 4], and map squares them.

Q47

Sorting Dictionaries by Value What is the correct way to sort a dictionary by its values in descending order?

```
data = {'a': 3, 'b': 1, 'c': 2}
```

A. `sorted(data, key=lambda x: x[1], reverse=True)`

B. `sorted(data.items(), key=lambda x: x[1], reverse=True)`

C. `sorted(data.items(), reverse=True)`

D. `sorted(data.items(), key=lambda x: x[0], reverse=True)`

Answer: (b) `sorted(data.items(), key=lambda x: x[1], reverse=True)`

Explanation: Sorting by values requires sorting `data.items()` with a key function on the second element of each pair.

Q48

Dictionary Comprehensions What does the following code output?

```
nums = [1, 2, 3]
squared = {n: n**2 for n in nums if n % 2 == 1}
print(squared)
```

A. {1: 1, 2: 4, 3: 9}

B. {1: 1, 3: 9}

C. {2: 4}

D. {}

Answer: (b) {1: 1, 3: 9}

Explanation: The comprehension filters odd numbers and maps them to their squares.

Q49

Understanding zip What does the following code output?

```
a = [1, 2, 3]
b = [4, 5]
zipped = list(zip(a, b))
print(zipped)
```

A. [(1, 4), (2, 5), (3, None)]

B. [(1, 4), (2, 5)]

C. [(4, 1), (5, 2)]

D. Raises an error

Answer: (b) [(1, 4), (2, 5)]

Explanation: zip stops when the shortest iterable is exhausted.

Q50

itertools Usage What does the following code output?

```
from itertools import permutations
perm = list(permutations([1, 2, 3], 2))
print(perm)
```

- A. [(1, 2), (2, 3)]
- B. [(1, 2), (1, 3), (2, 3)]
- C. [(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)]
- D. [(1, 2), (2, 1), (3, 2)]

Answer: (c) [(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)]

Explanation: permutations generates all ordered pairs of length 2 from the input list.

Answer Key Summary

The answer key below is formatted for quick checking after the detailed question-wise solutions.

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	B	11	B	21	B	31	B	41	B
2	B	12	A	22	B	32	B	42	B
3	B	13	B	23	B	33	A	43	A
4	B	14	C	24	A	34	C	44	A
5	C	15	A	25	C	35	A	45	A
6	A	16	B	26	D	36	B	46	B
7	C	17	B	27	B	37	B	47	B
8	B	18	A	28	A	38	B	48	B
9	C	19	A	29	B	39	B	49	B
10	C	20	A	30	B	40	B	50	C

