IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

Machine Learning Augmented Histopathological Diagnosis of an Aggressive Variant of Liver Cancer

Vivasvat Rastogi, Shashvat Rastogi

Vasant Valley School, Vasant Kunj Rd, Pocket 7, Sector C, Vasant Kunj, New Delhi, Delhi 110070, India; vivasvatrastogi@gmail.com
Mentor: Dr. Archana Rastogi

ABSTRACT: Liver cancer (Hepatocellular carcinoma) is one of the most common and fatal cancers. "Macrotrabecular-massive Hepatocellular carcinoma" (MTM-HCC), a recently identified aggressive variant, is characterized by thick trabeculae in 50% of the tumor area. Histopathology is the gold standard for diagnosis; however, microscopy is labor-intensive and time-consuming.

Four thousand and four hundred anonymized digitized histopathology images of HCC were used to create three main models and were randomly allocated to training (1600), validation (400), and test (400) for each. Convolutional neural network (CNN) model was created in Python using TensorFlow and Keras to predict the occurrence of Macrotrabecular patterns. After experimenting with different architectures (Inception V3, MobileNet V2, and ConvNet, VGG), optimizers and loss functions, Inception V3, RMSProp, and Binary Cross entropy were chosen. Performances were evaluated by confusion matrix and AUROC. Data augmentation improved accuracy by 25%. ML model-based Mobile App, for real-time detection of images, was developed. Independent internal and external validation was performed.

Training, validation, and test accuracies for Macrotrabecular vs non- Macrotrabecular and other patterns ranged from 96%-98%. AUROC ranged from 0.97-0.99. Internal and external validation showed an accuracy of 96-98%.

This model showed good performance and high accuracy. This is the first study to apply ML models to assist Pathologists in detecting an aggressive variant of liver cancer.

KEYWORDS: Computational biology and bioinformatics; Computational modeling; Machine learning (ML)-based mobile app; liver cancer; macrotrabecular -massive HCC.

Introduction

Liver cancer is the fifth most frequent and the second most fatal cancer worldwide.[1] Hepatocellular carcinoma (HCC) accounts for 85-90% of all malignant liver cancers.[2] HCC is associated with poor outcomes due to a lack of methods for early detection, accurate prognostication, and personalized treatments.[3]

Tumor tissue analysis has a well-established role in the diagnosis, theragnostic, and prognosis of HCC. Morphologic variants and phenotypes in tumor biopsies/ surgical specimens are important surrogate markers for molecular aberrations. Identifying various subtypes is a pre-requisite for prognostication, individualizing treatment, and correct allocation to new clinical trials.[3] Histopathology is the gold standard for the diagnosis of HCC subtypes.[3]

The macrotrabecular pattern of cancer cell arrangement (> six cells thick) represents an aggressive HCC subtype. Its presence in more than 50% of total tumors is classified as "Macrotrabecular -massive HCC (MTM-HCC)".[4] Hallmark features of this subtype are presentation with advanced clinical stage, large size, high tumor marker levels, vessel invasion, higher recurrence, reduced survival, and specific molecular derangements.[1,2] MTM-HCC needs more aggressive management, including liver transplantation and specific molecular therapies; hence, prompt and correct diagnosis is crucial.

Histopathological diagnosis of MTM-HCC requires manual microscopic examination of approximately 10-30 tumor tissue sections (with 1000s of fields/ patches).[5] This is a highly strenuous, time-consuming, and cumbersome task. Given the high tumor burden, there is an immediate need to adapt to newer methods of computational pathology. Digitization of tumor tissue slides and artificial intelligence (AI)-based analysis can impart objectivity, increase accuracy, and reduce inter-observer variability, which is a big step toward personalized medicine.

AI-based diagnostic and predictive liver cancer models are uncommon and created using radiology imaging. Data on AI-assisted Pathology image modeling is rare. The scarce literature on AI modeling on HCC pathology images includes a model for determining HCC differentiation, a model for distinguishing HCC from another less common type of liver cancer, and a model for predicting survival.[6-11] Machine Learning (ML) models can help in the quantification & early diagnosis of HCC subtypes. None of the published studies has evaluated AI's role in diagnosing aggressive MTM-HCC subtypes to assist Pathologists.

We hypothesize that ML models, created using digital histopathology images and their usage through a mobile app. can assist Pathologists in detecting MTM- HCC in large volumes of digitized images with high accuracy & objectivity and in a shorter time, thus helping in prognostication & timely decision of treatments. Therefore, the present work is aimed to diagnose an aggressive variant of liver cancer (MTM-HCC) by ML-assisted digital Pathology image analysis. The main objectives were to develop CNN models after experimenting

© 2023 Terra Science and Education

DOI: 10.36838/v5i6.1

with different architectures and to create a Mobile App that uses the ML model for real-time detection of Macrotrabecular patterns in digitized tissue slide images.

Methods

Data:

Digital images of hematoxylin & eosin-stained (HE) and formalin-fixed paraffin-embedded (FFPE) obtained from a pool of patients with HCC (n=105), who underwent surgical resection (partial or total hepatectomy) at a large liver dedicated tertiary care institute, were used. The images were retrieved from the study of histological patterns and their clinical relevance already published by the mentor.[5] This study required only archived digitized images with complete anonymity of the patient details. Thus, completely de-identified digital images were used and randomly numbered.

Four thousand four hundred digital images (200x magnification) representing different histological patterns of HCC, along with adjacent non-neoplastic liver and other primary cancer (cholangiocarcinoma), were used to develop the CNN model. The main histological patterns of HCC selected for the model were defined according to the published literature.[4,12] Macrotrabecular was defined as the trabecular thickness of more than six cancer cells; micro trabecular had a trabecular thickness of 3–5 cell thick plate; pseudoglandular pattern was characterized by trabeculae forming glandular lumina with or without bile or proteinaceous material in the lumina. **(Figure 1)**

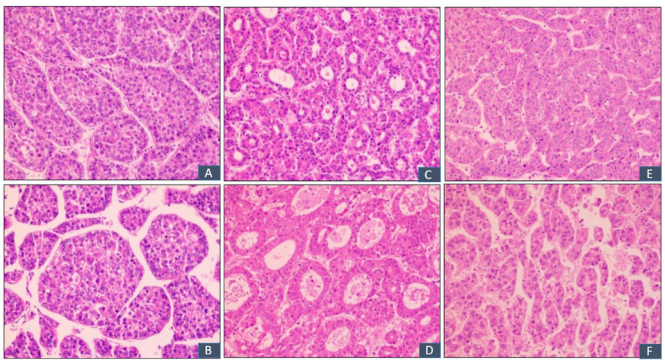

Figure 1: Histopathological images of macrotrabecular (A, B); pseudoglandular (C, D); microtrabecular (E, F) patterns. [HE stains; 200x magnification] Anonymized digital images were used for the ML model development.

Procedure:

Digital Histopathology images were scaled down to 25% of the original size, with a final size of 1024 x 822 pixels (from 40.4 MB to 3.4 MB).

CNN models were created, using supervised learning, in TensorFlow & Keras to distinguish Macrotrabecular from other patterns (Microtrabecular & Pseudoglandular). The model was programmed in Python. Details of the model development are provided later.

Accuracies of models were tested for four architectures (Inception V3, MobileNet V2, ConvNet, VGG), three optimizers, two loss functions, and with & without Data augmentation. Inception V3, RMSProp & Binary Cross entropy were chosen for the final training of models. Data augmentation improved accuracy by ~25%. Transfer Learning was applied for Inception V3.

A total of 6 models were created (3 main + 3 pre-test): Macrotrabecular vs. Microtrabecular, Macrotrabecular vs. Pseudoglandular, and Macrotrabecular vs. Non-Macrotrabecular, and one pre-test model for each (e.g., macrotrabecular+microtrabecular vs. others (normal, cholangiocarcinoma, pseudoglandular) for the 1st main model). To improve the generalisability of models, images of tissue with pre-analytical lab-related artifacts such as tissue folds, cracks, stain deposits, and over and under-stained sections were randomly mixed while modeling. The models were exported as tflite files. **(Figure 2)**

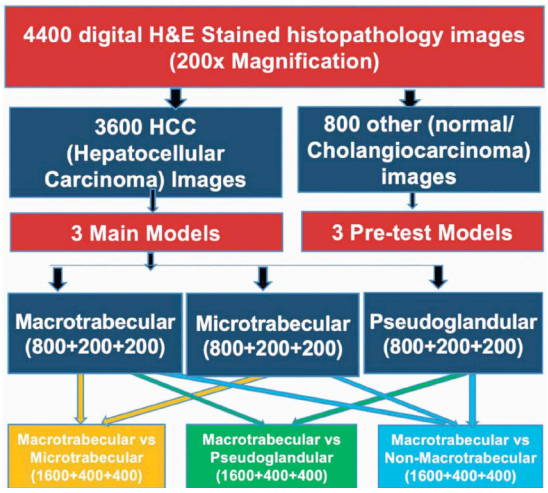

Figure 2: Data split for the creation of various models.

A mobile app was created with Flutter and Dart. The app (Livpath-AI) gives the option of 3 models to users, takes an image from the camera/gallery as input, runs the pre-test model, and provides prediction by running the main model if the pre-test was successful. **(Figure 3)**

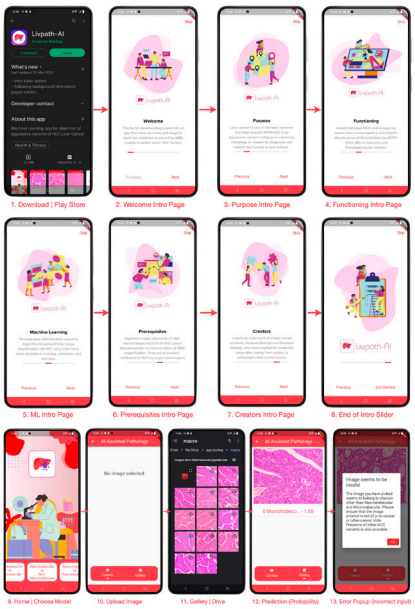

Figure 3: Livpath-AI Mobile App Workflow. Digital images acquired from the camera/gallery are analyzed by ML models, and the mobile application outputs the class probabilities.

Confusion matrices and AUROC curves were created based on test data predictions for the three main models. i.e., Macrotrabecular vs. Microtrabecular, Macrotrabecular vs. Pseudoglandular, and Macrotrabecular vs. non-Macrotrabecular.

Internal validation was performed on the independent test set 2 with HE images of FFPE cancer tissues captured at 400x magnification. External validation was performed by expert liver Pathologists of Public & Private hospitals by testing deidentified anonymized digital images of HE stained and 200x magnification using the Livpath-AI app.

Convolutional neural network (ML Model):

The model was programmed in Python using Tensorflow and Keras. Libraries used to carry out the task are shown below. The data, in the form of a zip file, was extracted from Google Drive onto Google Colaboratory (Figure 4).

```
import zipfile
import tensorflow as tf
from keras_preprocessing.image import ImageDataGenerator
from google.colab import drive

drive.mount('/content/drive')
local_zip = '/content/drive/MyDrive/Copy of train.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('tmp/train')
zip_ref.close()

local_zip = '/content/drive/MyDrive/Copy of valid.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('tmp/test')
zip_ref.close()
```

Figure 4: Data import.

The following code was written to carry out Data Augmentation, that is, reshaping the same image by rotating, zooming, shifting, etc., to increase accuracy with a relatively small dataset (Figure 5).

```
TRAINING_DIR = "tmp/train/train"
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest'
      )

VALIDATION_DIR = "tmp/test/valid"
validation_datagen = ImageDataGenerator(rescale = 1./255)

train_generator = training_datagen.flow_from_directory(
  TRAINING_DIR,
  target_size=(150,150),
  class_mode='categorical',
)

validation_generator = validation_datagen.flow_from_directory(
  VALIDATION_DIR,
  target_size=(150,150),
  class_mode='categorical',
)
```

Figure 5: Data augmentation.

CNN Models with the following layers/architecture were prepared: InceptionV3 using transfer learning, MobileNet V2 from Tensorflow, ConvNet with a Dense final layer for the binary prediction task, and VGG (Visual Geometry Group). These were used independently and not in the same notebook (Figures 6-9).

```
#InceptionV3
!wget --no-check-certificate \
    https://storage.googleapis.com/mledu-
datasets/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5 \
    -O /tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5

from tensorflow.keras.applications.inception_v3 import InceptionV3
local_weights_file = '/tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'
pre_trained_model = InceptionV3(input_shape = (150, 150, 3),
include_top = False,
weights = None)
pre_trained_model.load_weights(local_weights_file)

for layer in pre_trained_model.layers:
  layer.trainable = False
last_layer = pre_trained_model.get_layer('mixed7')
last_output = last_layer.output
x = layers.Flatten()(last_output)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.2)(x)
x = layers.Dense  (2, activation='softmax')(x)

model = Model(pre_trained_model.input, x)
```

Figure 6: InceptionV3.

```
#VGG
model = tf.keras.models.Sequential([
      # BLOCK-1
          tf.keras.layers.Conv2D(32, (2,2), activation = 'relu',
input_shape=(150,150,3), padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Conv2D(32, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.MaxPool2D((2,2)),
          tf.keras.layers.Dropout(0.2),
      # BLOCK-2
          tf.keras.layers.Conv2D(64, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Conv2D(64, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.MaxPool2D((2,2)),
          tf.keras.layers.Dropout(0.3),
      # BLOCK-3
          tf.keras.layers.Conv2D(128, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Conv2D(128, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.MaxPool2D((2,2)),
          tf.keras.layers.Dropout(0.4),
      # BLOCK-4
          tf.keras.layers.Conv2D(256, (2,2), activation = 'relu', padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Conv2D(256, (2,2), activation = 'relu',  padding = 'same'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.MaxPool2D((2,2)),
          tf.keras.layers.Dropout(0.4),
      # End BLOCK          tf.keras.layers.Flatten(),
          tf.keras.layers.Dense(512, activation = 'relu'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Dropout(0.5),
          tf.keras.layers.Dense(256, activation = 'relu'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Dropout(0.3),
          tf.keras.layers.Dense(512, activation = 'relu'),
          tf.keras.layers.BatchNormalization(),
          tf.keras.layers.Dropout(0.3),
          tf.keras.layers.Dense(2, activation='softmax')
 ])
```

Figure 7: VGG.

```
#ConvNet
model = tf.keras.models.Sequential([
   tf.keras.layers.Conv2D(64, (2,2), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
   tf.keras.layers.Conv2D(64, (2,2), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
   tf.keras.layers.Conv2D(128, (2,2), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
   tf.keras.layers.Conv2D(128, (2,2), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
   tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
   tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
```

Figure 8: ConvNet.

```
#MobileNetV2
mobile_net_layers = tf.keras.applications.mobilenet_v2.MobileNetV2(
    input_shape=None,
    alpha=1.0,
    include_top=True,
    weights='imagenet',
    input_tensor=None,
    pooling=None,
    classes=1000,
    classifier_activation='softmax'
)
mobile_net_layers.trainable = False
model = tf.keras.Sequential([
  mobile_net_layers,
  tf.keras.layers.Dropout(0.3),
  tf.keras.layers.Dense(2,activation='softmax')
])
```

Figure 9: MobileNetV2.

The model was compiled using several optimizers and loss functions (one combination at a time) (Figure 10).

```
model.compile(loss = 'binary_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])
```

Figure 10: Model compilation.

The model was trained by defining the training, validation, epochs, and call back dataset (Figure 11).

```
history = model.fit(train_generator, epochs=256, validation_data =
validation_generator, verbose = 1, callbacks=[callbacks])
```

Figure 11: Training.

DOI: 10.36838/v5i6.1

The model was exported as a .tflite file. A mobile app with Flutter, based on Dart Programming language, was further created and published on Google Play Store (https://play.google.com/store/apps/details?id=com.vivasvatrastogi.livpathai).

A link to the GitHub repository is as follows: https://github.com/Vivasvat-Rastogi/flutterML

Results and Discussion

Performance Evaluation of Models:

Initial training and validation were performed with different combinations of architectures, optimizers, loss functions, and with or without data augmentation. Accuracies are shown in Table 1.

Table 1: Initial experimentation accuracies – InceptionV3, Binary crossentropy, and RMSprop showed highest accuracies.

Criteria	Divisions	Training Accuracy	Validation Accuracy
Data Augmentation	With Data Augmentation	0.82	0.95
	Without Data Augmentation	0.61	0.68
Optimisers	rmsprop	0.82	0.95
	Adam	0.83	0.90
	SGD	0.75	0.85
Loss Functions	binary_crossentropy	0.82	0.95
	Mean_squared_error	0.84	0.85
Architectures	ConvNet	0.82	0.95
	VGG	0.93	0.75
	Inception V3	0.95	0.98
	MobileNet V2	0.82	0.97

Based on the above, all the models were trained with InceptionV3 architecture using transfer learning, rmsprop optimizer, binary cross entropy loss function, and data augmentation. The final main models' accuracies were in the range of 97.5-98.5 and are shown in Table 2.

Table 2: Final models' accuracies ranged from 97.5 to 98.5.

Model	Training / Validation/ Test Accuracy		
Macro- vs Micro-trabecular	98	97.5	97.5
Macro- vs Pseudoglandular	98.5	97.5	98
Macro- vs non-macrotrabecular	98.5	98	98.5

The below confusion matrices show the actual vs. model-predicted data statistics. (Figure 12)

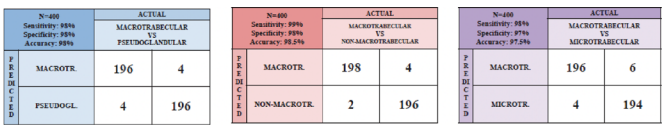

Figure 12: Confusion Matrices. Accuracy for different models 96-98%

Training, validation, and test accuracies ranged from 96%-98%. The area under ROC for different test groups ranged from 0.97-0.99. (Figure 13)

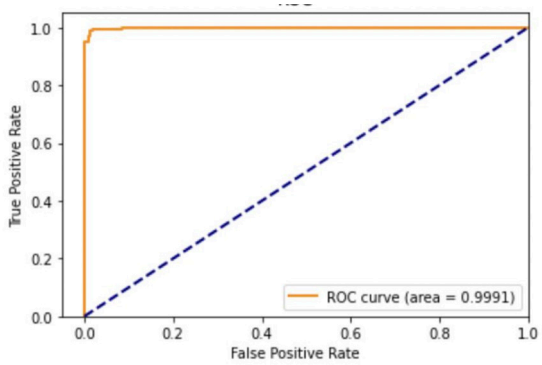

Figure 13: AUROC for Macrotrabecular versus non-Macrotrabecular was 0.99

Internal validation in an independent set of images at higher magnification (400x) displayed an accuracy of 98%. Whereas, Livpath-AI-based external validation by two different expert pathologists showed an accuracy of 96%.

Discussion

In the present study, a machine learning model and a mobile App were developed to detect an aggressive variant of liver cancer using digital histopathology images with high accuracy and objectivity in a very short time. Given the high tumor burden, scarcity of Pathologists, and concerns about the time & labor of expert Pathologists, this model has great significance. Once HCC is diagnosed by Pathologists, Machine learning (ML) based detection of Macrotrabecular patterns can assist them by reducing their workload & saving time. Even lab staff or junior pathologists can capture and test digital images with the mobile App, and expert pathologists can verify the model result remotely. Thus, in the present study, the integration of computational pathology & artificial intelligence was done to assist histopathologists.

HCC subtypes have distinct morphological characteristics and unique clinical and biological correlates. These subtypes are extremely relevant to prognosis and therapeutic decisions. Recent publications have highlighted the importance of tissue diagnosis, especially for the subtypes of HCC in the era of molecular therapies.[1,3,5]

Ziol *et al.* identified a histologic subtype of HCC associated with specific molecular features and poor prognostic factors such as large tumor size, high serum alpha-fetoprotein (AFP) levels, satellite nodules, and vascular invasion. The authors designated this subtype as "MTM-HCC." This subtype was an independent predictor of early and overall recurrence.[1]

These studies developed CNN models based on histological images and displayed high accuracy. In the study by Zeng *et al.*, areas under the receiver operating characteristic curves (AUCs) for prediction ranged from 0.78-0.91. Their different models generalized well in the validation dataset with AUCs ranging from 0.81- 0.92.[11] Cheng *et al.* model showed good performance with an AUC value of 93.5%.[10] Lin *et al.* demonstrated over 90% accuracy of their CNN model for determining HCC differentiation.

Conclusion

Summary:

Four thousand four hundred anonymized digital images of HCC tumor tissue were randomly segregated for the train-

ing, validation, and test groups in a ratio of 4:1:1. Patch size for model development was 1024x822 pixels. CNN model was created in Python; TensorFlow and Keras were applied to distinguish macrotrabecular architecture from microtrabecular and pseudoglandular. Different architectures (ConvNet, VGG, Inception V3, Mobile Net V2), optimizers & loss functions were experimented with and finally Inception V3 (transfer learning), RMSProp, and Binary Cross entropy were chosen. Very high accuracy was achieved with InceptionV3 and MobileNet V2. Confusion Matrix and AUROC showed high accuracy (96-96%) for internal and external validation data sets. In addition, Mobile App (Android and iOS) was created using Flutter & Dart.

In conclusion:

ML (CNN) models created using digitized histopathology images of HCC showed good performance and high accuracy in detecting the MTM-HCC subtype. This model is the first to apply ML to assist Pathologists in diagnosing aggressive variants of liver cancer. ML models & Livpath-AI app quantifying the digital histopathological features can make Pathologist's tasks less labor intensive and time efficient. Early diagnosis with highly accurate ML assistance to Pathologists is a step forward toward optimizing patient care.

The way forward is testing and validating on a large cohort of HCC images from different centers and applying other CNN architectures.

Acknowledgments

The authors thank Dr. Archana Rastogi, Professor (Department of Pathology), Institute of Liver & Biliary Sciences, Delhi, India, for providing anonymized digitized histopathological images of liver cancer.

References

1. Ziol, M.; Poté, N.; Amaddeo, G.; Laurent, A.; Nault, J. C.; Oberti F.; Costentin, C.; Michalak, S.; Bouattour, M.; Francoz, C.; Pageaux, G. P.; Ramos, J.; Decaens, T.; Luciani, A.; Guiu, B.; Vilgrain, V.; Aubé, C.; Derman, J.; Charpy, C.; Zucman-Rossi, J.; Barget, N.; Seror, O.; Ganne-Carrié, N.; Paradis, V.; Calderaro, J. Macrotrabecular-massive hepatocellular carcinoma: A distinctive histological subtype with clinical relevance. Hepatology 2018, Jul,68 (1),103-112. Doi: 10.1002/hep.29762. Epub 2018 May 9. PMID:29281854.
2. Li, X.; Yao, Q.; Liu, C.; Wang, J.; Zhang, H.; Li, S; Cai, P. Macrotrabecular-Massive Hepatocellular Carcinoma: What Should We Know? *J Hepatocell Carcinoma* **2022**, *5*(9), 379-387. Doi: 10.2147/JHC.S364742. PMID: 35547829; PMCID: PMC9084381.
3. Rastogi, A. Changing role of histopathology in the diagnosis and management of hepatocellular carcinoma. *World J Gastroenterol* **2018,** *24*(35), 4000-4013. Doi: 10.3748/wjg.v24.i35.4000. PMID: 30254404; PMCID: PMC6148422.
4. Calderaro, J.; Couchy, G.; Imbeaud, S.; Amaddeo, G.; Letouze, E.; Blanc, J.F.; Laurent, C.; Hajji, Y.; Azoulay, D.; Bioulac-Sage, P.; Nault, J.C.; Zucman-Rossi, J. Histological subtypes of hepatocellular carcinoma are related to gene mutations and molecular tumor classification. *J. Hepatol.* **2017**, *4,* 727–738.
5. Rastogi, A.; Maiwall, R.; Ramakrishna, G.; Modi, S.; Taneja, K.; Bihari, C.; Kumar, G.; Patil, N.; Thapar, S.; Choudhury, A.K.; Mukund, A.; Pamecha, V.; Sarin, S.K. Hepatocellular carcinoma: Clinicopathologic associations amidst marked phenotypic heterogeneity. *Pathol Res Pract*. **2021**, *217*,153290. Doi: 10.1016/j. prp.2020.153290. Epub 2020 Nov 24. PMID: 33307344.
6. Kiani, A.; Uyumazturk, B.; Rajpurkar, P.; Wang, A.; Gao, R.; Jones, E.; Yu, Y.; Langlotz, C.P.; Ball, R.L.; Montine, T.J.; Martin, B.A.; Berry, G.J.; Ozawa, M.G.; Hazard, F.K.; Brown, R.A.; Chen, S.B.; Wood, M.; Allard, L.S.; Ylagan, L.; Ng, A.Y.; Shen, J. Impact of a deep learning assistant on the histopathologic classification of liver cancer. *NPJ Digit Med* **2020**, *3* (23). PMID: 32140566 Doi: 10.1038/s41746-020-0232-8.
7. Saillard, C.; Schmauch, B.; Laifa, O.; Moarii, M.; Toldo, S.; Zaslavskiy, M.; Pronier, E.; Laurent, A.; Amaddeo, G.; Regnault, H.; Sommacale, D.; Ziol, M.; Pawlotsky, J.M.; Mulé, S.; Luciani, A.; Wainrib, G.; Clozel, T.; Courtiol, P.; Calderaro, J. Predicting Survival After Hepatocellular Carcinoma Resection Using Deep Learning on Histological Slides. Hepatology 2020, 72 (2000-2013). PMID: 32108950 Doi: 10.1002/hep.31207.
8. Lin, H.; Wei, C.; Wang, G.; Chen, H.; Lin, L.; Ni, M.; Chen, J.; Zhuo, S. Automated classification of hepatocellular carcinoma differentiation using multiphoton microscopy and deep learning. *J Biophotonics* **2019**, *12*(7), e201800435. PMID: 30868728 Doi: 10.1002/jbio.201800435.
9. Ahn, J. C.; Qureshi, T. A.; Singal, A. G.; Li, D.; Yang, J. D. Deep learning in hepatocellular carcinoma: Current status and future perspectives. *World J Hepatol*. **2021**, *13*(12),2039-2051. Doi: 10.4254/wjh.v13.i12.2039. PMID: 35070007; PMCID: PMC8727204.
10. Cheng, N.; Ren, Y.; Zhou, J.; Zhang, Y.; Wang, D.; Zhang, X.; Chen, B.; Liu, F.; Lv, J.; Cao, Q.; Chen, S.; Du, H.; Hui, D.; Weng, Z.; Liang, Q.; Su, B.; Tang, L.; Han, L.; Chen, J.; Shao, C. Deep Learning-Based Classification of Hepatocellular Nodular Lesions on Whole-Slide Histopathologic Images. *Gastroenterology* **2022**, *162* (7):1948-1961.e7. Doi: 10.1053/j.gastro.2022.02.025. Epub 2022 Feb 22. PMID: 35202643.
11. Zeng, Q.; Klein, C.; Caruso, S.; Maille, P.; Laleh, N. G.; Sommacale, D.; Laurent, A.; Amaddeo, G.; Gentien, D.; Rapinat, A.; Regnault, H.; Charpy, C.; Nguyen, C.T.; Tournigand, C.; Brustia, R.; Pawlotsky, J.M.; Kather, J.N.; Maiuri, M.C.; Loménie, N.; Calderaro, J. Artificial intelligence predicts immune and inflammatory gene signatures directly from hepatocellular carcinoma histology. *J Hepatol*. **2022**,*77*(1):116-127. Doi: 10.1016/j. jhep.2022.01.018. Epub 2022 Feb 7. PMID: 35143898.
12. Tan, P. S.; Nakagawa, S; Goossens, N.; Venkatesh, A.; Huang, T.; Ward, S. C.; Sun, X.; Song, W.M.; Koh, A.; Canasto-Chibuque,C.; Deshmukh, M.; Nair, V.; Mahajan, M.; Zhang, B.; Fiel, M.I.; Kobayashi, M.; Kumada, H.; Hoshida, Y. Clinico pathological indices to predict hepatocellular carcinoma molecular classification. *Liver Int.* **2016**, *36* (1), 108–118.
13. Liao, H.; Long, Y.; Han, R.; Wang, W.; Xu, L.; Liao, M.; Zhang, Z.; Wu, Z.; Shang, X.; Li, X.; Peng, J.; Yuan, K.; Zeng, Y. Deep learning-based classification and mutation prediction from histopathological images of hepatocellular carcinoma. *Clin Transl Med* **2020**, *10*, e102. PMID: 32536036 Doi: 10.1002/ctm2.102
14. Wang, H.; Jiang, Y.; Li, B.; Cui, Y.; Li, D.; Li, R. Single-Cell Spatial Analysis of Tumor and Immune Microenvironment on Whole-Slide Image Reveals Hepatocellular Carcinoma Subtypes. *Cancers (Basel)* **2020**, *12* (12), 3562. Doi: 10.3390/cancers12123562.PMID: 33260561; PMCID: PMC7761227.
15. Chen, M.; Zhang, B.; Topatana, W.; Cao, J.; Zhu, H.; Jueng panich, S.; Mao, Q.; Yu, H.; Cai, X. Classification and Mutation Prediction based on Histopathology H&E images in Liver Cancer using Deep Learning. *NPJ Precis Oncol* **2020**, *4*, 14. PMID: 32550270 Doi: 10.1038/s41698-020-0120-3.
16. Lu, L.; Daigle, B. J. Jr. Prognostic Analysis of Histopathological Images using Pre-trained Convolutional Neural Networks: Applica

DOI: 10.36838/v5i6.1

tion to Hepatocellular carcinoma. *PeerJ* **2020**, *8*, e8668. Doi: 10.7717/peerj.8668. PMID: 32201640; PMCID: PMC7073245.
17. Shi, J. Y.; Wang, X.; Ding, G. Y.; Dong, Z.; Han, J.; Guan, Z.; Ma, L. J.; Zheng, Y.; Zhang, L.; Yu, G. Z.; Wang, X. Y.; Ding, Z. B.; Ke, A. W.; Yang, H.; Wang, L.; Ai, L.; Cao, Y.; Zhou, J.; Fan, J.; Liu, X.; Gao, Q. Exploring prognostic indicators in the pathological images of hepatocellular carcinoma based on deep learning. *Gut* **2021**,*70* (5), 951-961. Doi: 10.1136/gutjnl-2020-320930. PMID: 32998878 Doi: 10.1136/gutjnl-2020-320930.
18. Yamashita, R.; Long, J.; Saleem, A.; Rubin, D. L.; Shen, J. Deep Learning Predicts Postsurgical Recurrence of Hepatocellular Carcinoma from Digital Histopathologic Images. *Sci Rep* **2021**, *11*(1), 2047. Doi: 10.1038/s41598-021-81506-y. PMID: 33479370; PMCID: PMC7820423

Authors

Vivasvat Rastogi and Shashvat Rastogi are brothers and seniors at Vasant Valley School, Delhi, India.

Both authors are keenly interested in technology, Computer Science, Math, Music, and Sanskrit. They have won several prestigious STEM awards at international, national, state, and school levels and play several musical instruments. They presented this research work in Regeneron ISEF 2023 under the category "Computational Biology & Bioinformatics" and won the fourth grand award in this category.