Decision Analytics for Business & Policy

Monkeypox Detection

Project Report

I. Link to Source Code & Data

II. Resources:

1) **Data Sources:** 1) Monkeypox Skin Lesion dataset on Kaggle; 2) Monkeypox cases in the U.S. available at Centers for Disease Control and Prevention (CDC)
2) CNN Architecture: Kaggle.

3) **Literature**:

Ali, S. N., Ahmed, M. T., Paul, J., Jahan, T., Sani, S. M. Sakeef, Noor, N., & Hasan, T. (2022). Monkeypox Skin Lesion Detection Using Deep Learning Models: A Preliminary Feasibility Study. Arxiv Preprint, Cornell University.

Betti, Matthew, Lauren Farrell, and Jane M. Heffernan (2022). "A pair formation model with recovery: Application to monkeypox." medRxiv.

Bhunu C, Garira W, Magombedze G (2009). Mathematical analysis of a two-strain HIV/aids model with antiretroviral treatment. Acta Biotheor 57(3):361–381

Bhunu C, Mushayabasa S (2011). Modeling the transmission dynamics of pox-like infections. IAENG Int J 41(2):1–9

Grant R, Nguyen LL, Breban R. Modelling human-to-human transmission of monkeypox. Bull World Health Organ. 2020 Sep 1;98(9):638-640.

Forese, Henri (2020). Infectious Disease Modeling: Fit Your Model to Coronavirus Data. Available at https://tinyurl.com/mr2drtya. Towards Data Science. Accessed on November 24, 2022.

Odom, Mary R., R. Curtis Hendrickson, and Elliot J. Lefkowitz (2009). "Poxvirus protein evolution: family-wide assessment of possible horizontal gene transfer events." Virus research 144, no. 1-2 : 233-249.

Peter, Olumuyiwa James, Sumit Kumar, Nitu Kumari, Festus Abiodun Oguntolu, Kayode Oshinubi, and Rabiu Musa (2022). "Transmission dynamics of Monkeypox virus: a mathematical modeling approach." Modeling Earth Systems and Environment 8, no. 3 : 3423-3434.

III. Team Member Contributions:

Tasks	**By**
Project Proposal	All
Project Data & Model Plan	All
Cleaning the dataset	All
Training and Testing the Dataset	Ricardo, Yu
Validating Performance	Ricardo, Yu
Creating a front-end app to gather inputs (optional)	Yu
Writing a report	All
Preparing Presentation	Bikash, Mahrukh

IV. Problem Statement

Monkeypox is a skin disease that was first detected in humans in 1970. The disease has two strains: Clade I and Clade II. The first strain is highly virulent and has a fatality of over 10%. The second strain is less fatal than the first, and 99 in 100 infected will survive it. In 2022, the second strain circulated to 100 countries and spread to over 80,000 people globally, including 30,000 in the United States (CDC). The symptoms of the monkeypox virus usually include rashes on or around private parts, skin eruption, fever, and headache among others(WHO).

Combatting Monkeypox is an urgent issue, especially against the backdrop of more than two years of the COVID pandemic. While not considered as transmissible and as deadly as COVID-19, Monkeypox carries with it the disease fatigue experienced by the wider world. More particularly, because the disease is reported to be found mostly among male homosexuals, some communities such as LGBTQ face the wrath of dehumanizing episodes of discrimination, reminiscent of those during the early AIDS days. However, monkeypox is not limited to any particular population or ethnic group and everyone in the population is equally susceptible to contracting the virus.

Monkeypox lesions look similar to other skin lesions which makes it difficult to diagnose. Furthermore, because rashes are common on or around private parts in Monkeypox, the disease can be confused with other sexually transmitted diseases (STDs), creating high chances of early misdiagnosis. This delay in diagnosis can allow the diseases to spread unnoticed, making it difficult for the government to execute timely health policy protocols.

We propose an app that exploits neural networks to identify monkeypox from other skin infections and allows people to self-diagnose their condition. This intervention will allow the government health departments to plan out isolation and vaccination protocols quickly in areas of high prevalence.

V. Data Summary

We have used the Monkeypox Skin Lesion dataset from Kaggle. The dataset contains two sub-data folders that have 228 original images and 3192 augmented images of the skin infection for the cases of monkeypox, chickenpox, and measles. Each image is approximately the size of 5 to 9 KB. We will use the images in the augmented dataset to train a neural network model and use it to predict the presence of monkeypox disease on test data in the original image datasets.

For the SIR model, we have referred to input parameters from existing literature in this area. Key parameters include beta (total probability of infection for a susceptible person), gamma (probability of recovery for an infected person), and r-naught (basic reproductive number). Prior studies have used different r-naught values such as 2.3 (Betti et al. 2022), 2.13 with uncertainty bounds 1.46 - 2.67 (Grant et al. 2020); a beta value of 0.00025 for rodent-to-human contact rate, 0.00006 for human-to-human contact rate, 0.027 for rodent-to-rodent contact rate (Bhunu 2011); gamma value of 0.83 for humans recovery rate (Bhunu 2009) and 0.07 (for 2-weeks of recovery) according to WHO; and a delta value of 0.5 (Peter et al. 2022), 0.2 (Odom et al. 2009), 0.04% according to CDC.

In addition, we have used data on the daily case counts for monkeypox from the Centers for Disease Control and Prevention (CDC) for data on the United States. The data variables are:

Variable	Description
epi_date_V2	Date on which case was reported
Cases	Total cases per day
7-day average	7-day moving average of cases
Cumulative Cases	Total cumulative cases

VI. Analytical Formulation

1) SIR Model

We calibrated a SIR model following the dynamic version (changing R naught) of the framework reviewed in class, and with the following extensions:

1. We included a third category of population: Deceased (D). This category will allow us to differentiate the population that has recovered from the population that has died due to the disease. This addition to the model is of interest to determine the social costs of the disease, and, ultimately, for policy making.
2. To accommodate the new category 'Deceased', we included a third and fourth path of transition. Besides 'Susceptible' to 'Infected' and 'Infected' to 'Recovered,' those 'Infected' can transition directly into 'Death' and those 'Susceptible' can transition directly into 'Recovered' if vaccinated.
3. We also consider a parameter v to account for the population that is being vaccinated in each period. Those vaccinated transition from Susceptible to recovered without having to be Infected.
4. The parameter q captures the proportion of people that isolates, so they remove themselves from the interaction between Susceptible and Infected.
5. We added equation 4 which will give the trajectory of the mortality rate due to the monkeypox disease.
6. Finally we added equation 5 as a helper function to use for fitting the model to the data.

The equations of the model are the following.

$$\frac{\delta S_t}{\delta t} = -\delta S_t - \frac{\beta(1-q)}{N} I_t S_t - v S_t \quad (1)$$

$$\frac{\delta I_t}{\delta t} = \frac{\beta(1-q)}{N} I_t S_t - I_t(\gamma + \theta) \quad (2)$$

$$\frac{\delta R_t}{\delta t} = \gamma I_t + v S_t \quad (3)$$

$$\frac{\delta D_t}{\delta t} = \theta I_t \quad (4)$$

$$\frac{\delta I2_t}{\delta t} = \frac{\beta(1-q)}{N} I_t S_t \quad (5)$$

We calibrated the model using data on daily cases of monkeypox in the United States because it recorded the highest outbreaks among the counties for which data on daily case count is available.

2) Neural Network Model

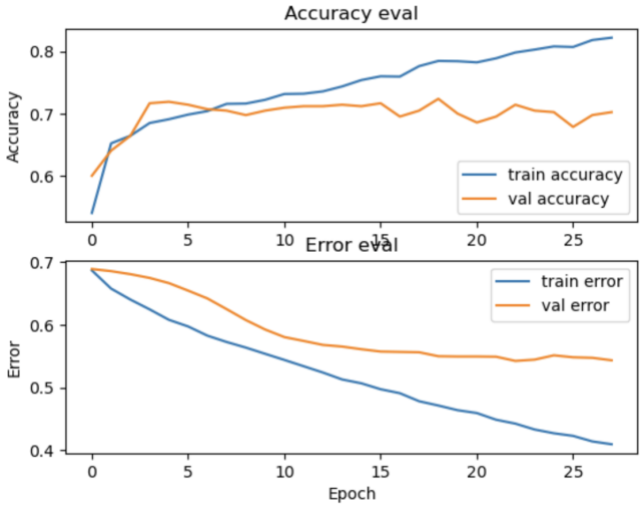
We stored the training data (2142 images), validation data (420 images), and test data (45 images) in separate files and used keras.preprocessing.image.ImageDataGenerator.flow_from_directory() to transform the images into arrays. The dimensions of the training/validation/test arrays are (n, 224, 224, 3), where n is the number of images in each set, 224 stands for the width and height of the images, and 3 denotes RGB (red, green, blue) channels.

We used the convolutional neural network architecture from the Kaggle community. The architecture consists of 5 convolutional layers, each of them respectively has 32, 64, 32, 128, and 512 filters, and uses ReLU as an activation function. The output of the convolutional layers is then fed into 2 dense layers, each respectively using ReLU and Sigmoid as activation functions, and consisting of 32 and 1 nodes. The Sigmoid activation function ensures that the output is between 0 and 1 and offers us the probability of an image as Monkeypox or not.

The convolutional neural network is trained with the Tensorflow library using Adam as the optimizer and the learning rate of 0.000005. Then, we used the test set to check the accuracy of the model we trained.

The accuracy and the error for each epoch show that our model is learning the data, with a little sign of overfitting as the difference between train accuracy and validation accuracy becomes larger toward the end of the training. The test accuracy of our model is 73%.

Figure 1: CNN Model Training and Validation Accuracy for each epoch



The confusion matrix (Figure 1) shows the performance of our classifier. We ran the model on a test set of 45 images that it had not previously seen, 20 of which were positives (Monkeypox) and 25 were negatives (non-Monkeypox). The threshold for classifying an observation as monkeypox was set to 0.4, as we prefer having negative people diagnosed as positives more than having positives falsely diagnosed as non-Monkeypox. With that threshold, we obtained 33 correct classifications (73% test accuracy), 11 false positives (24%), and 1 false negative (2%). The model's True Positive and False Negative Rates can be seen using the ROC curve as shown in Figure 2 for different levels of thresholds.

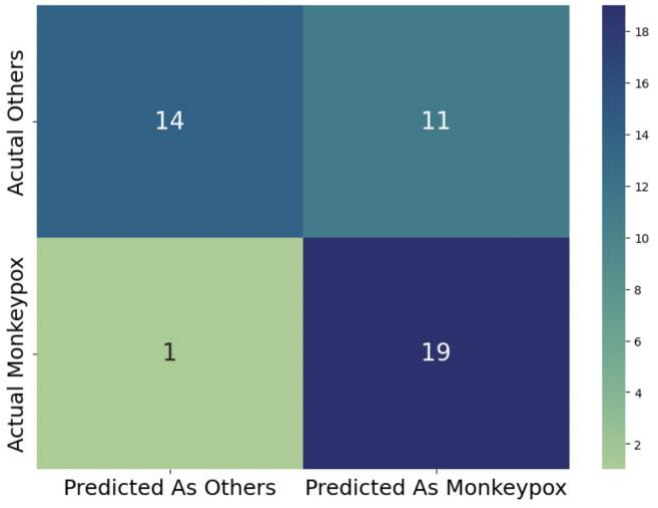


Figure 2: CNN Confusion Matrix

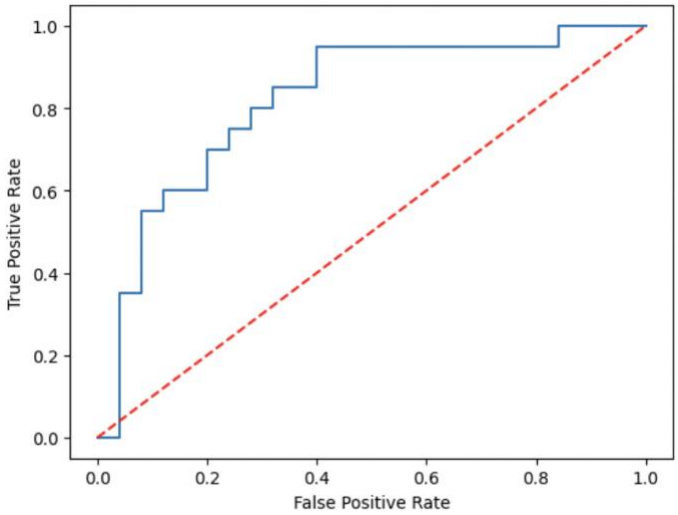
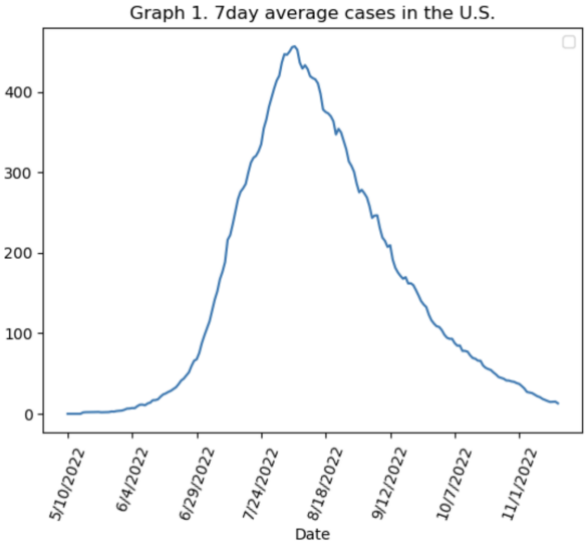


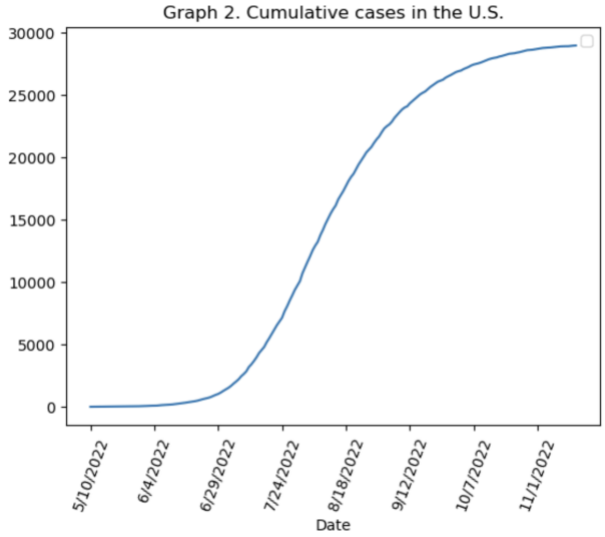
Figure 3: CNN ROC Curve

VII. Implementation & Analysis

SIR model

Graph 1 shows the 7-day moving average of Monkeypox daily cases (positive tests) registered in the U.S., while Graph 2 depicts the cumulative cases during the 6-month period that goes from May 5th, 2022 to November 16th, 2022. Daily cases in the U.S. reached a maximum of 637 cases on 08/01/2022. By the end of the period, when the outbreak seems to have dwindled, the total cumulative cases reached 28,995.





As a high-level explanation of the procedure followed, we took the dynamic SIR model and obtained the best parameters of the dynamic R0 by fitting the data of cumulative cases to the helper function I2 (equation 5, described previously in this document). After knowing how the dynamic R0 behaves, now we can access the S, I, R, and D curves.

Going into a more detailed explanation, the first step was to set some of the parameters from the literature (gamma, theta, q, v, and population), and then translate our SIR(D) model from the equations into code. These parameters come from the literature: 18 is a reasonable estimate for the time it takes a person to fully recover from monkeypox and stop being contagious; we needed to adjust slightly the death rate for the model to output a number of cumulative deaths close to the actual number in the data. The strain that spread in the U.S. had a substantially lower death rate than other strains previously studied, so we set the death rate to 0.005%. The vaccination rate was calculated as the total vaccination (approx one million) over the number of days during which vaccines were administered (approx. six months) times the population. Lastly, the population considered (N total) was adjusted to only include in "Susceptibles" the total population from the states where 65% of the total disease was concentrated, this was 144 million people. For the setup of the SIR model and the definition of the dynamic R0, we borrowed and adjusted the extended version of the SIR code learned in class.

```
# PARAMETERS
gamma = (1/18)    # this is 0.0555
theta = 0.005/100
q = 0.02
vacc = 1000000/(300000000 * 6 * 30) # this is 0.0000185
N_total = 144321749
N = 144321749
```

0.2s

```
# SIR MODEL
def deriv_dynamic(y, t, beta, gamma, theta, q, vacc, N):
    S, I, R, D, I2 = y
    dSdt = -beta(t) * (1 - q) * S * I / N - vacc * S
    dIdt = beta(t) * (1 - q) * S * I / N - (gamma + theta) * I
    dRdt = gamma * I + vacc * S
    dDdt = theta * I
    # I2 is the cumulative cases curve:
    dI2dt = beta(t) * (1 - q) * S * I / N
    return dSdt, dIdt, dRdt, dDdt, dI2dt
```

0.1s

The second step consisted in defining the dynamic R naught as a function of the parameters to be computed during the fitting process (R0_start, R0_end, k and x0, where x0 is the date where R0 changes at the fastest pace, and k determines how fast R0 changes).

Third, we then define the model that includes the definition of dynamic beta (gamma times dynamic R naught), we pass the initial conditions vector and call the *odeint* function to get the S, I, R, D, and I2 curves back, as well as the dynamic R0.

```
# Defining the dynamic R0 and the model
# In developing part of this code, we used part of the class SIR Extensions code
def logistic_R_0(t, R_0_start, k, x0, R_0_end):
  #x0 = lockdown date, k = how fast R0 changes
  return (R_0_start-R_0_end) / (1 + np.exp(-k*(-t+x0))) + R_0_end

# Defining our model
def Model(days, N_total, R_0_start, k, x0, R_0_end):
    def beta(t):
        return logistic_R_0(t, R_0_start, k, x0, R_0_end) * gamma

    # set all initial conditions: S0, I0, R0, D0, I20
    y0 = (N) - 5, 5, 0, 0, 5 # we start with five onfected only,
    t = np.linspace(0, days-1, days)
    ret = odeint(deriv_dynamic, y0, t, args=(beta, gamma, theta, q, vacc, N))
    S, I, R, D, I2 = ret.T
    R_0_over_time = [beta(i)/gamma for i in range(len(t))]

    return t, S, I, R, D, I2, R_0_over_time
```

✓ 0.3s

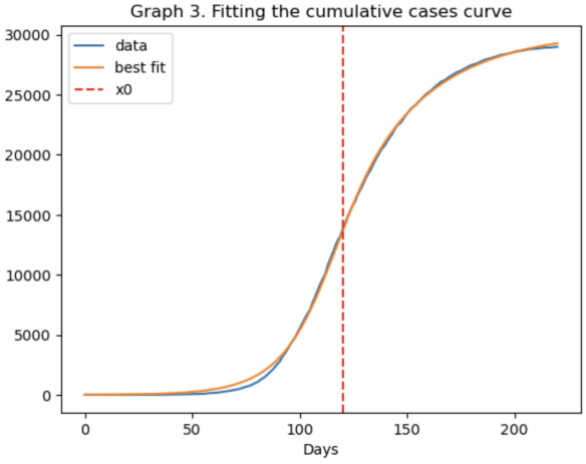
In fourth place, we defined a fitter function that calls the Model function described above and returns the I2 curve. With this fitter function, we contoured the cumulative cases data to the I2 curve and received the set of parameters that best adjust our SIR(D) model to the data through the dynamic beta. We used Scipy's *lmfit* for our curve fitting. The code required us to provide an initial guess and a max and min for the parameters to be fitted, and we used least squares to fit the cumulative cases (Graph 2). In developing this part of the code, we consulted and adapted part of the work of Froese, 2020 (see references).

```
# Use cumulative cases for the fitting
cases_usa = np.array(cases_usapd['Cumulative Cases'])

# In developing this part of the code, We used some of the ideas from here:
# https://towardsdatascience.com/infectious-disease-modelling-fit-your-model-to-coronavirus-data-2568e672dbc7
def fitter(x, R_0_start, k, x0, R_0_end):
    ret = Model(days, N_total, R_0_start, k, x0, R_0_end)
    return ret[5] # I2 curve

outbreak_day = 10   # Lets say the outbreak starts on day 10
data = cases_usa  # data
days = outbreak_day + len(data)
y_data = np.concatenate((np.zeros(outbreak_day), data))
x_data = np.linspace(0, days - 1, days, dtype=int)

params_init = {"R_0_start": (0.1, -5.000001, 10.0), "k": (0.1, -5.01, 8.0),
               "x0": (10, 0, 150), "R_0_end": (0.1, -5.00001, 5)}
               # {parameter: (initial guess, minimum value, max value)}
mod = lmfit.Model(fitter)
for j, (init, mini, maxi) in params_init.items():
    mod.set_param_hint(str(j), value=init, min=mini, max=maxi, vary=True)
params = mod.make_params()
fit_method = "leastsq"
result = mod.fit(y_data, params, method="least_squares", x=x_data)
```

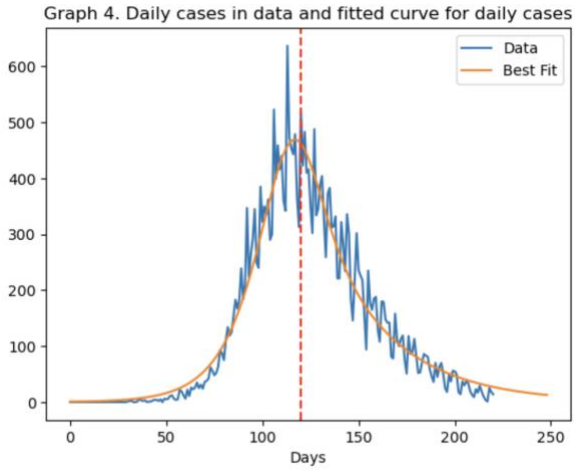
✓ 0.6s

The result of fitting the I2 curve of our model to the cumulative cases in the U.S. is shown in Graph 3, which includes x0 -the day when R naught changed the most- as a red vertical line. Graph 4 shows an overlapping of our model-fitted daily cases to the 7-day average data. As can be seen, x0 matches with the inflection point of the daily cases: as R0 decreases, the transmission of monkeypox slows down and eventually stops.

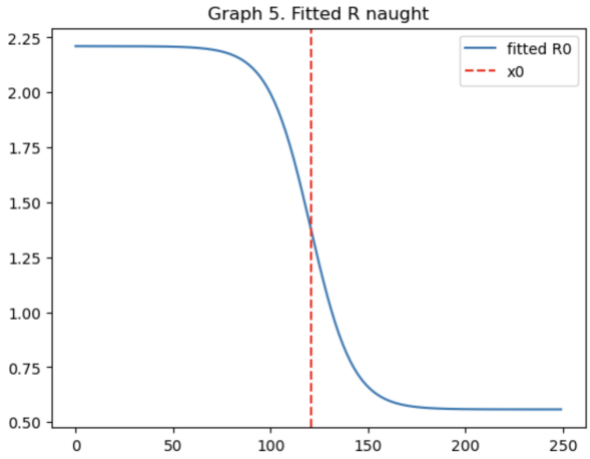
Graph 3. Fitting the cumulative cases curve

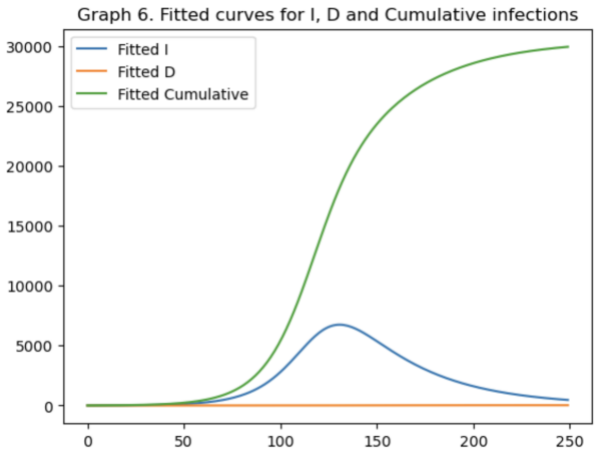


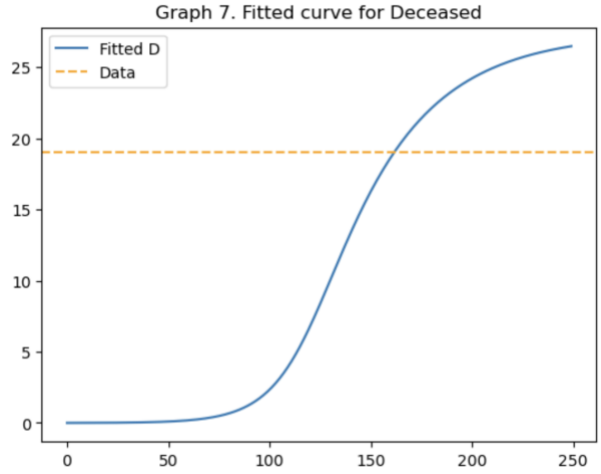
Graph 4. Daily cases in data and fitted curve for daily cases



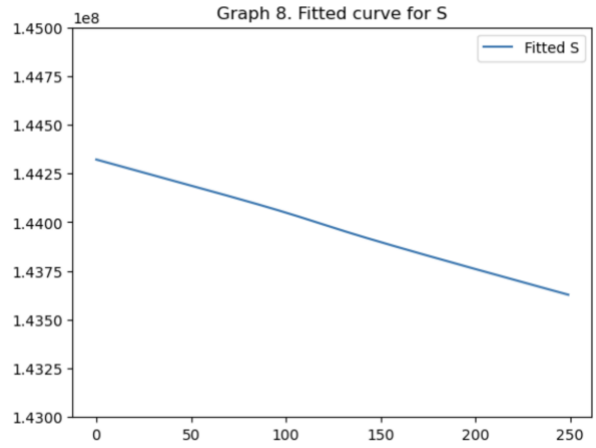
Graph 5 portrays the trajectory of R naught, which goes from 2.21 to 0.55, with the fastest-changing pace at day x0 (120). As can be inferred from the numbers presented so far, the severity of the monkeypox outbreak was nowhere near that of the COVID-19 pandemic; therefore, the difference in scales does not allow presenting the trajectories for Susceptibles, Infected, Recovered, and Deceased all in one graph. As a result, the fitted I, D, and cumulative cases (curve I2) trajectories are depicted in Graph 6: curve I reaches a peak on day 131 at 6,741, and starts to go down until reaching zero. On the other hand, the cumulative cases reach 29.913. Lastly, the fitted curve for D, also shown in Graph 7 for more clarity, has an S shape with its steepest slope on day 120 and stabilizes at 26. This number is a close match with the actual number of deaths in the data (19 deceased in the U.S.).







Lastly, the S-fitted curve decreases slowly, moving both to recoveries and deaths from infection, as well as from some vaccinations.



Streamlit App

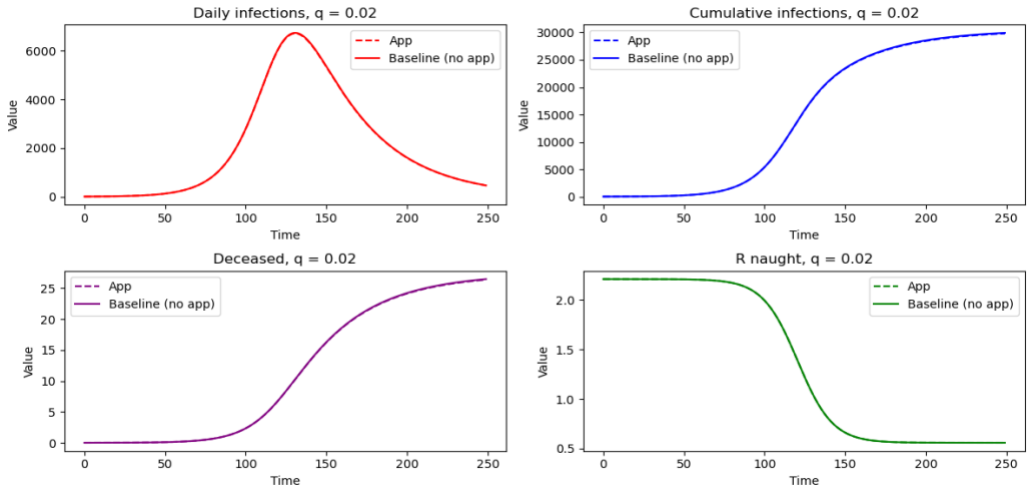
We used the Streamlit application package in Python to create an app, backed by the neural network, that can be used to detect monkeypox. The user can upload a picture of the skin rash and click "Upload" and the application will display the message "Monkeypox Detected" or "Monkeypox Not Detected".

Link Between Application & SIR Model

People will be able to stay at home, test for the monkeypox virus, and self-isolate. Once more and more people start to isolate at home this will directly impact the 'q' or isolation parameter in the SIR model. There will be less interaction between the Susceptible and Infected populations which will eventually contain the spread of the disease. This analysis builds on one key assumption that human beings are rational and they will self-isolate once they get a positive result from the application. To see how the rate of isolation will impact the transmission of the virus, we experimented with a few different scenarios:

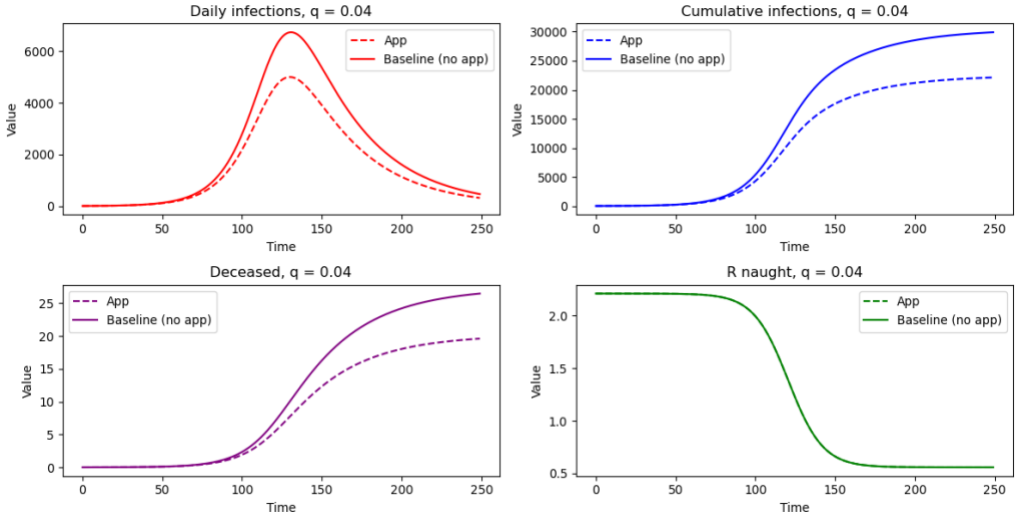
<u>Scenario-0: Baseline Isolation is 2%</u>

At baseline, the rate of isolation, q, is 2%. The daily infections, cumulative infections, deaths, and R-naught have the following trend. The model estimates a total of 26 deaths.



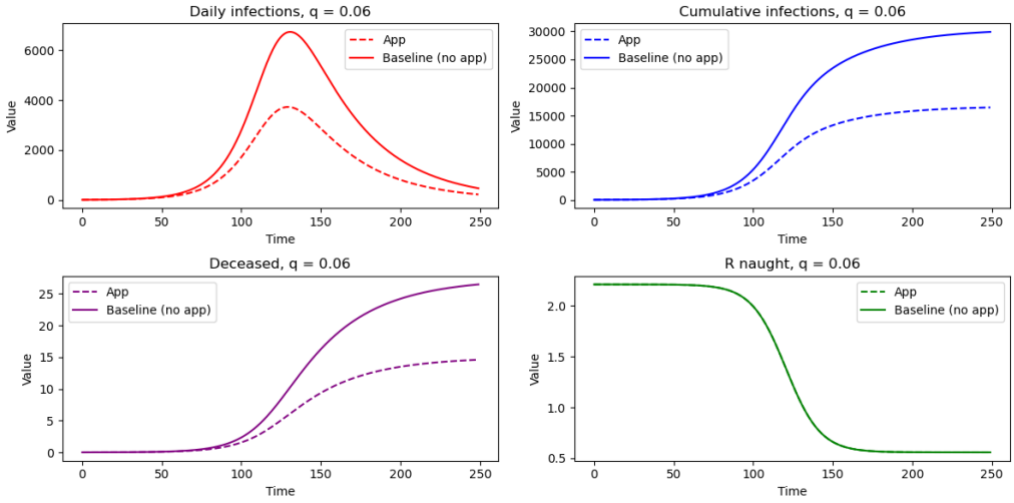
Scenario-1: Application Usage/Isolation is 4%

If we are able to double the number of people using the application, there will be a reduction in the infections and the total deaths are 19 i.e. we will be able to save 7 lives.



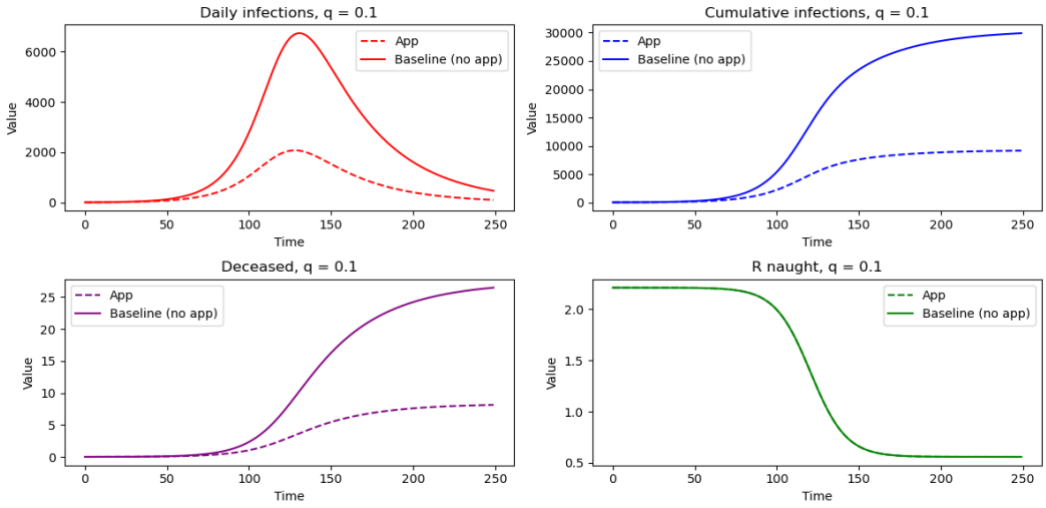
Scenario-2: Application Usage/Isolation is 6%

If the application usage and isolation is 6% then there will be a further reduction in infections and the total deaths will be 14 i.e. we will be able to save 12 lives.



Scenario-3: Application Usage/Isolation is 10%

If the application users increase by 5 times then there will be a further reduction in infections and the total deaths will be 8 i.e. we will be able to save 18 lives.



VIII. Policy Implications & Recommendations

The application, backed by the neural network model, will allow people to self-test and self-isolate. First, this will result in huge cost savings for the government because there will be no need to set up testing centers at various locations. The funds saved could be used elsewhere, such as for patient care in hospitals for critical patients or to roll out vaccination drives. Second, if the application is deployed by the government then the daily case count can come directly from the application; once a person is tested positive at home they will come 'in the system' and the government will be able to track daily cases more efficiently.

IX. Limitations

During this project, we encountered the following limitations.

1) Data limitations
    - Data related to deaths were not available. We could have used the death rate to fit the model as a robustness check.
    - The county-level data were only available for Los Angeles county whereas the state-level data were only available for California, New York, and Maryland. More granular data could help improve the model's accuracy.
    - Because Monkeypox is not as widely contagious, it leaves us with less data to track and model. It also impacts the SIR model implementation.

2) Model limitations
    - A 'good' model is a function of a large swathe of data. As mentioned, unlike COVID-19, the disease remained largely contained and there were not enough cases to produce as close of a model to reality.
    - SIR-specific limitation
        i) Monkeypox also infected a small (and scattered) proportion of the population. However, we incorporated an entire country's population when implementing the model.
    - Neural Network
        i) We only had 200 images to train our model. We worked around this limitation by augmenting the images by twisting, turning, and filtering them. However, more data would have helped strengthen the model.
    - In our SIR model, we took the population (N) for only those states that reported more than 500 monkeypox cases.

3) Policy Limitation:
    - In our model, we assumed that the people would act rationally and self-isolate once they are diagnosed with Monkeypox. However, because of the disease's striking resemblance with many STD symptoms, people may be hesitant to seek treatment or tests. Furthermore, socio-economic factors such as job compulsion may compel people to continue with their lives as usual despite having Monkeypox. If people do not follow further precautions, it may, therefore, be difficult to tame the disease. The decentralized app idea that we put forward may, therefore, affect the chances of curbing the disease.
    - The impact of isolation on the containment of monkeypox depends on the usage of the app which requires a smartphone, an internet connection, and the ability to use a smartphone.
    - We suspect that the use of the app for early and accurate detection would modify the other policy tools such as the vaccination rate.

X. Conclusion

A big factor in controlling the spread of disease in any population is how adaptable the system is towards reporting the cases, managing patient influx, and implementing a national policy to contain the virus. We attempted to solve this multi-dimensional policy decision by exploring ways to detect the virus quickly so that the government is in a better position to respond in case of a national emergency. We used the SIR model for monkeypox cases in the United States between May and November 2022 to predict the trend of the spread of the virus depending on key parameters including the rate of recovery, isolation, vaccination, and disease-induced death rate. We find that <add from policy recommendations>. Future research should explore ways to make the model more robust to small increases in daily case count relative to the population and should use data at a more granular level in population clusters where monkeypox is detected.